Fast Light Transport Analysis

Kevin Matzen

29 September 2011

Motivation

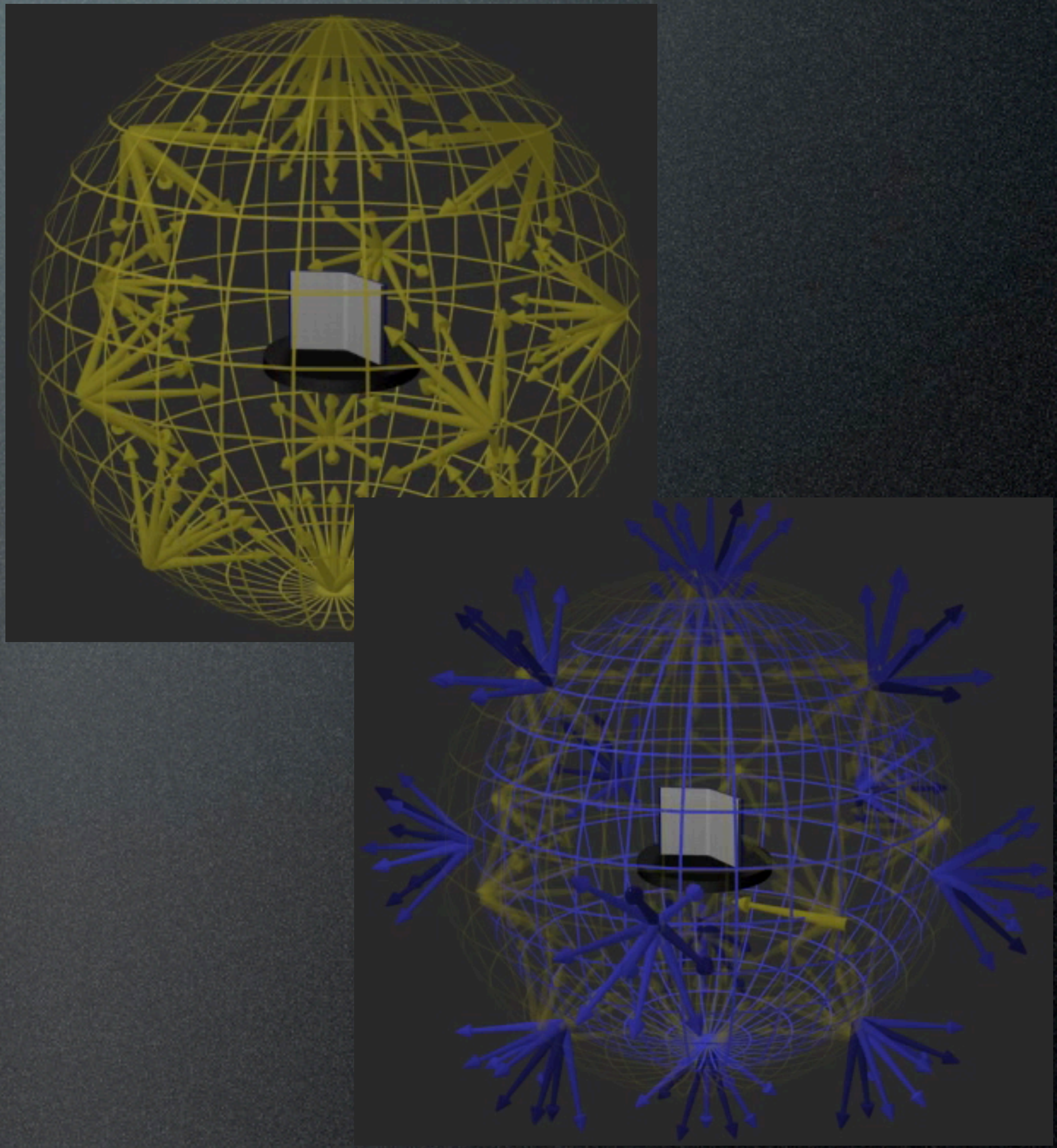
- How might you solve the following?
  - Relight scene with novel illumination
  - Render image from novel viewpoint
  - Extract scene's illumination

Methods we've seen

- Recover geometry
- Infer materials
- Render the scene
- Question: What if we have direct access to the scene at one point?
  - Fewer heuristic methods
  - Actual, physical measurements
  - What would we measure?

Light Transport

- Relation between incident and exitant light of static scene
- Linear
    - Simple linear photon model
    - No interference or diffraction
- Simply, light sums
- All bounces effectively summed



Transport Tensor

Exitant light

$$E = TI$$

4D tensor (E)

8D tensor (T)

4D tensor (I)

Incident light

Why 4D?

- Why not 5D or 6D?
- x, y, z, normal - 5 parameters
- Same anywhere along normal (up to a scale factor) - eliminates 1 parameter
- Consider surface of convex hull only - surface coordinates + normal - 4 parameters
- Consider both incident and exitant to get 4+4 = 8D total

Transport Matrix

Exitant light

$$E = TI$$

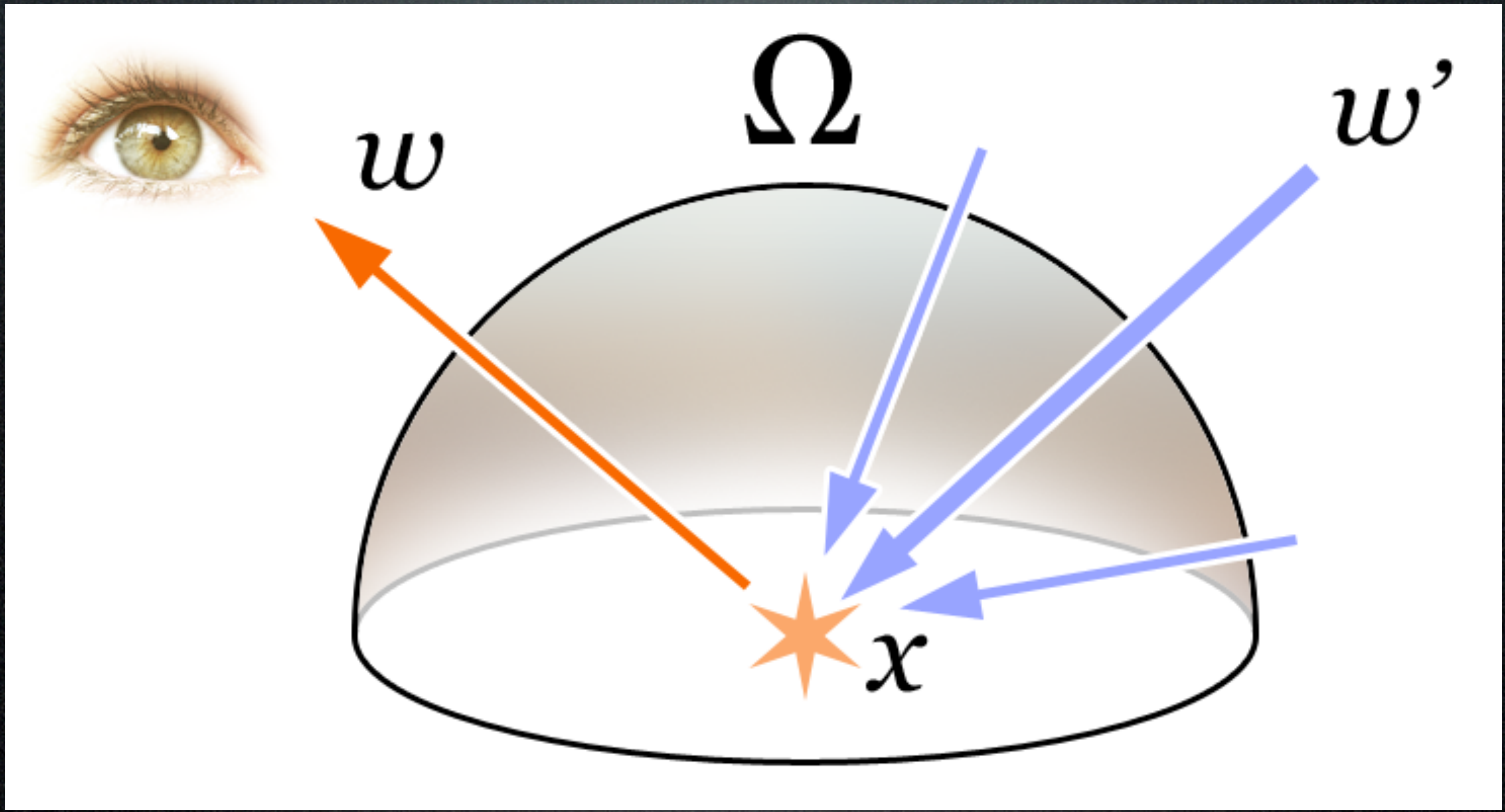
vector — matrix — vector

Incident light

For convenience, flatten out

Rendering Equation

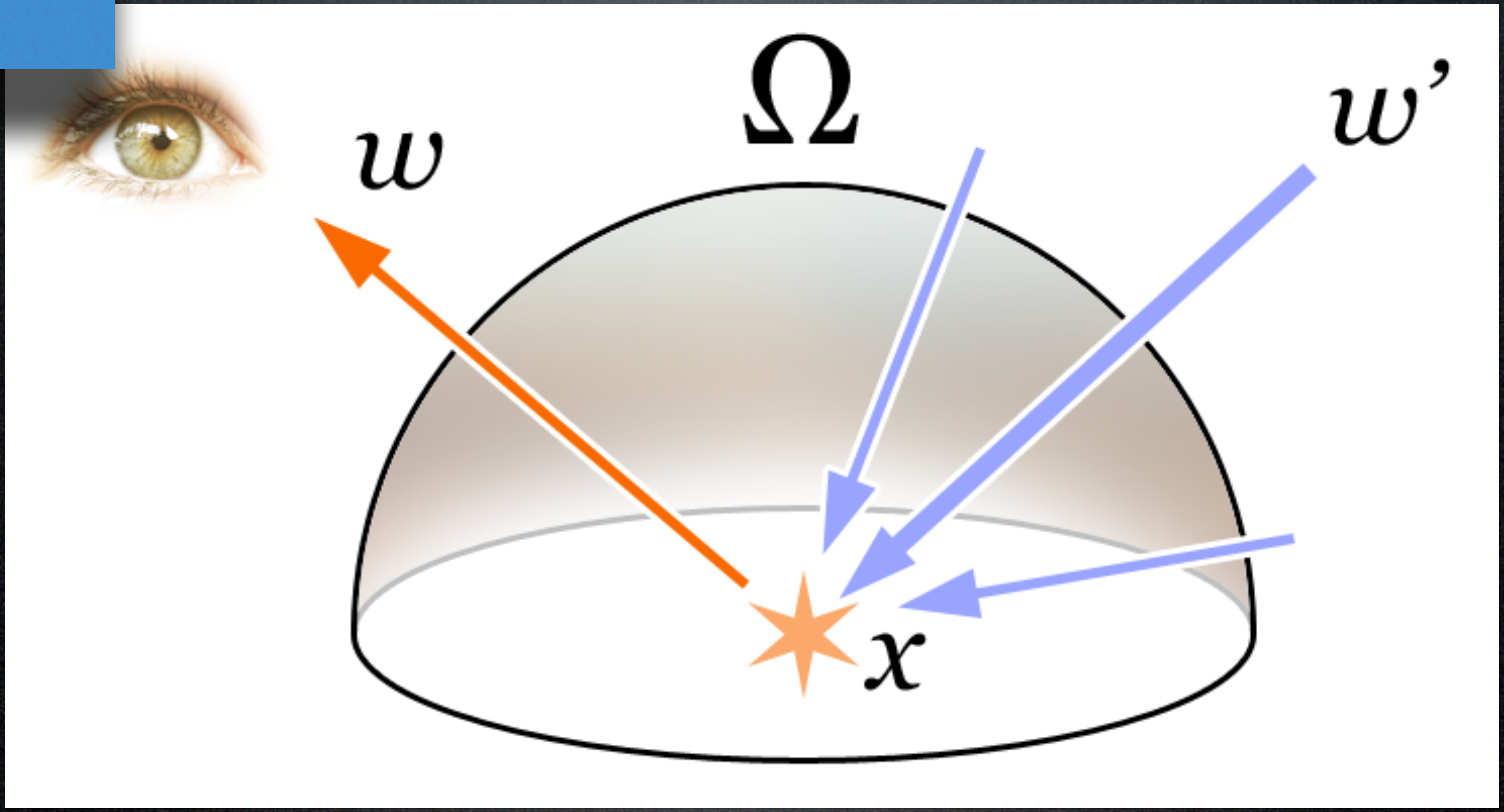
$$L_o(\mathbf{x}, \omega, \lambda, t) = L_e(\mathbf{x}, \omega, \lambda, t) + \int_\Omega f_r(\mathbf{x}, \omega', \omega, \lambda, t) L_i(\mathbf{x}, \omega', \lambda, t)(-\omega' \cdot \mathbf{n}) d\omega'$$



Rendering Equation

$$L_o(\mathbf{x},\omega,\lambda,t) = L_e(\mathbf{x},\omega,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega',\omega,\lambda,t) L_i(\mathbf{x},\omega',\lambda,t)(-\omega'\cdot\mathbf{n})d\omega'$$

Light out

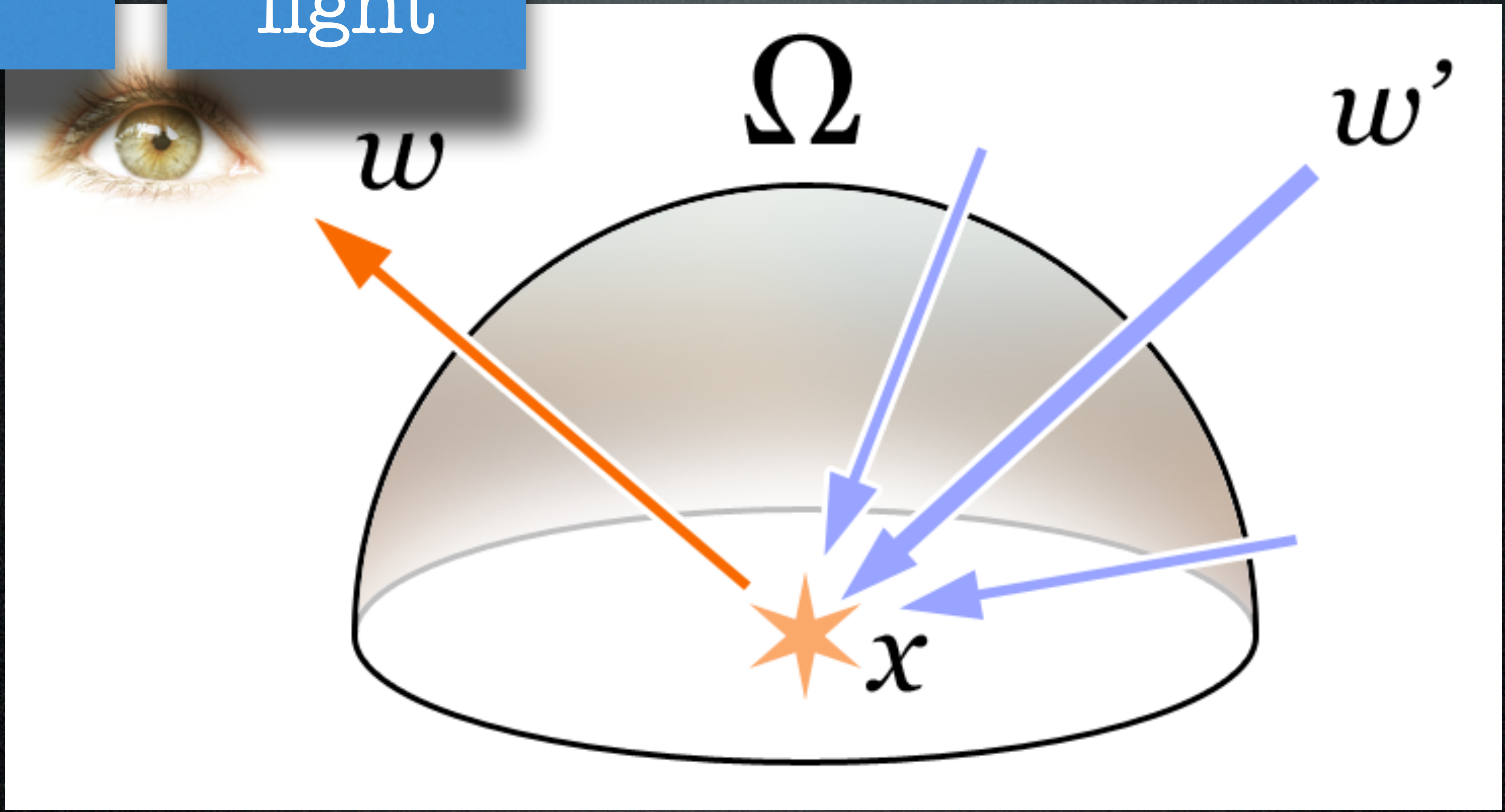


Rendering Equation

$$L_o(\mathbf{x},\omega,\lambda,t) = L_e(\mathbf{x},\omega,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega',\omega,\lambda,t) L_i(\mathbf{x},\omega',\lambda,t)(-\omega'\cdot\mathbf{n})d\omega'$$

Light out

Emitted light



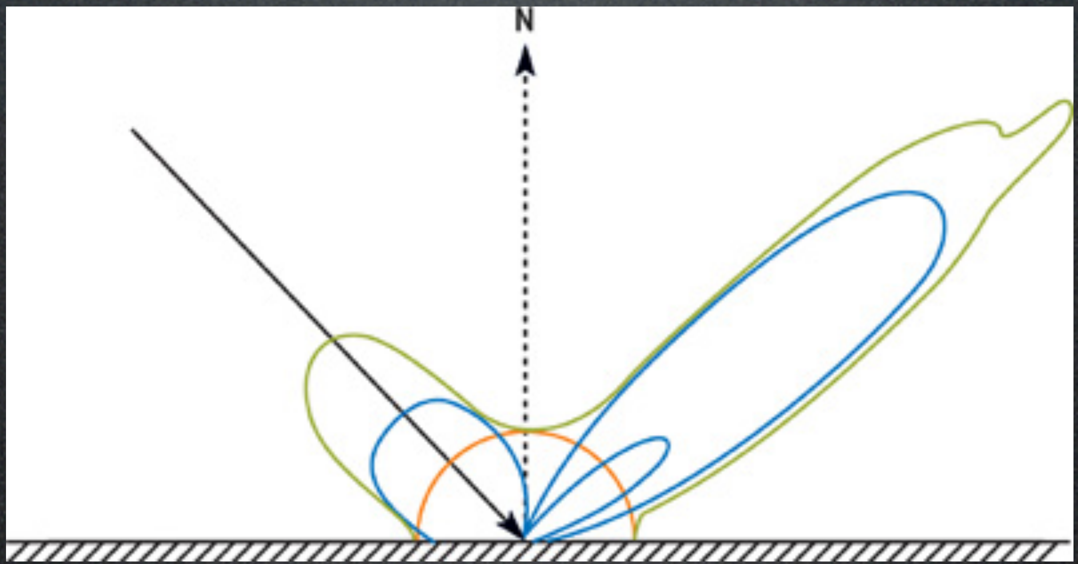
Rendering Equation

$$L_o(\mathbf{x},\omega,\lambda,t) = L_e(\mathbf{x},\omega,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega',\omega,\lambda,t) L_i(\mathbf{x},\omega',\lambda,t)(-\omega'\cdot\mathbf{n})d\omega'$$

Light out

Emitted light

BRDF



Rendering Equation

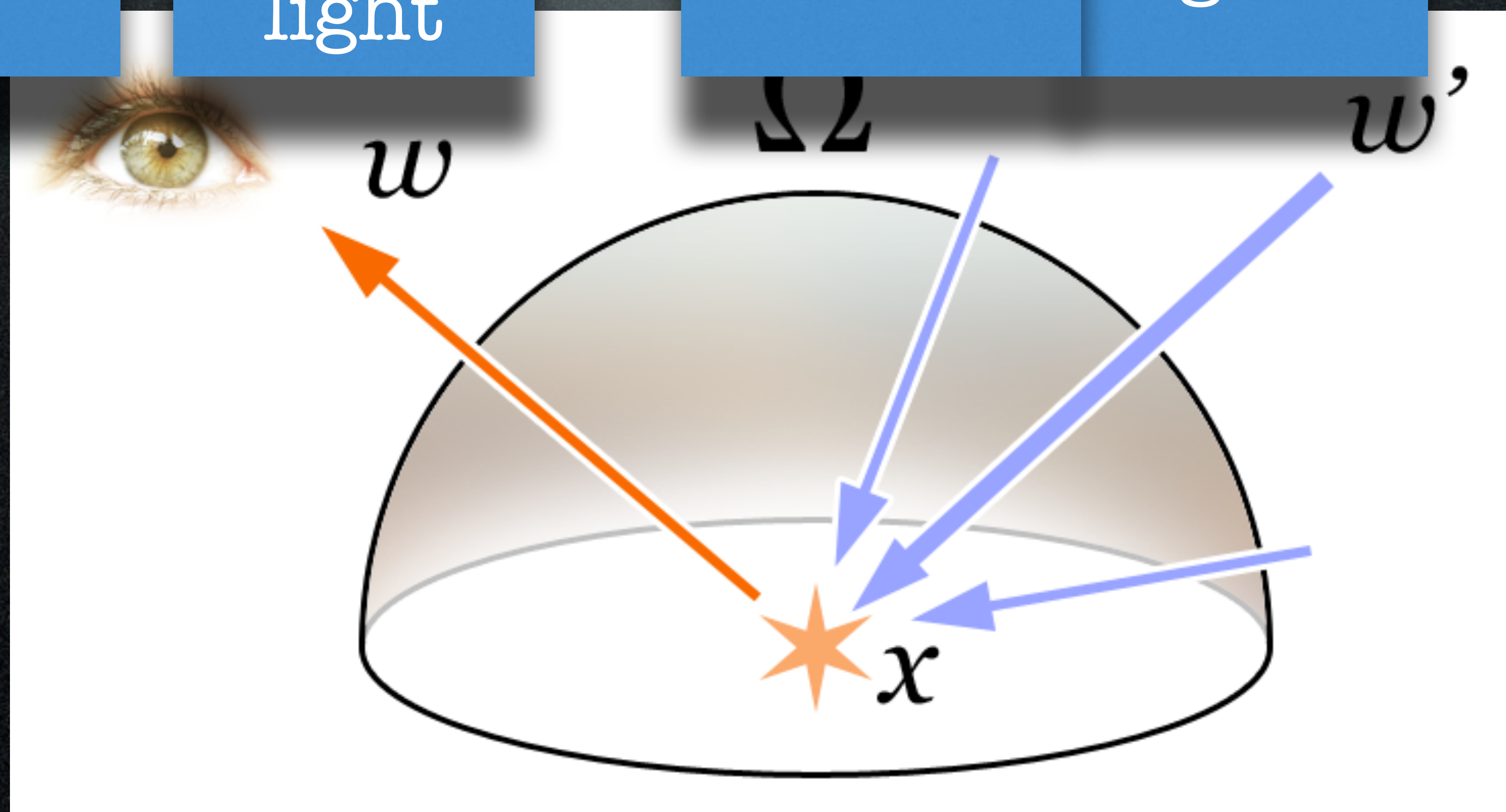
$$L_o(\mathbf{x},\omega,\lambda,t) = L_e(\mathbf{x},\omega,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega',\omega,\lambda,t) L_i(\mathbf{x},\omega',\lambda,t)(-\omega'\cdot\mathbf{n})d\omega'$$

Light out

Emitted light

BRDF

Light in



Rendering Equation

$$L_o(\mathbf{x},\omega,\lambda,t) = L_e(\mathbf{x},\omega,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega',\omega,\lambda,t)L_i(\mathbf{x},\omega',\lambda,t)(-\omega'\cdot\mathbf{n})d\omega'$$

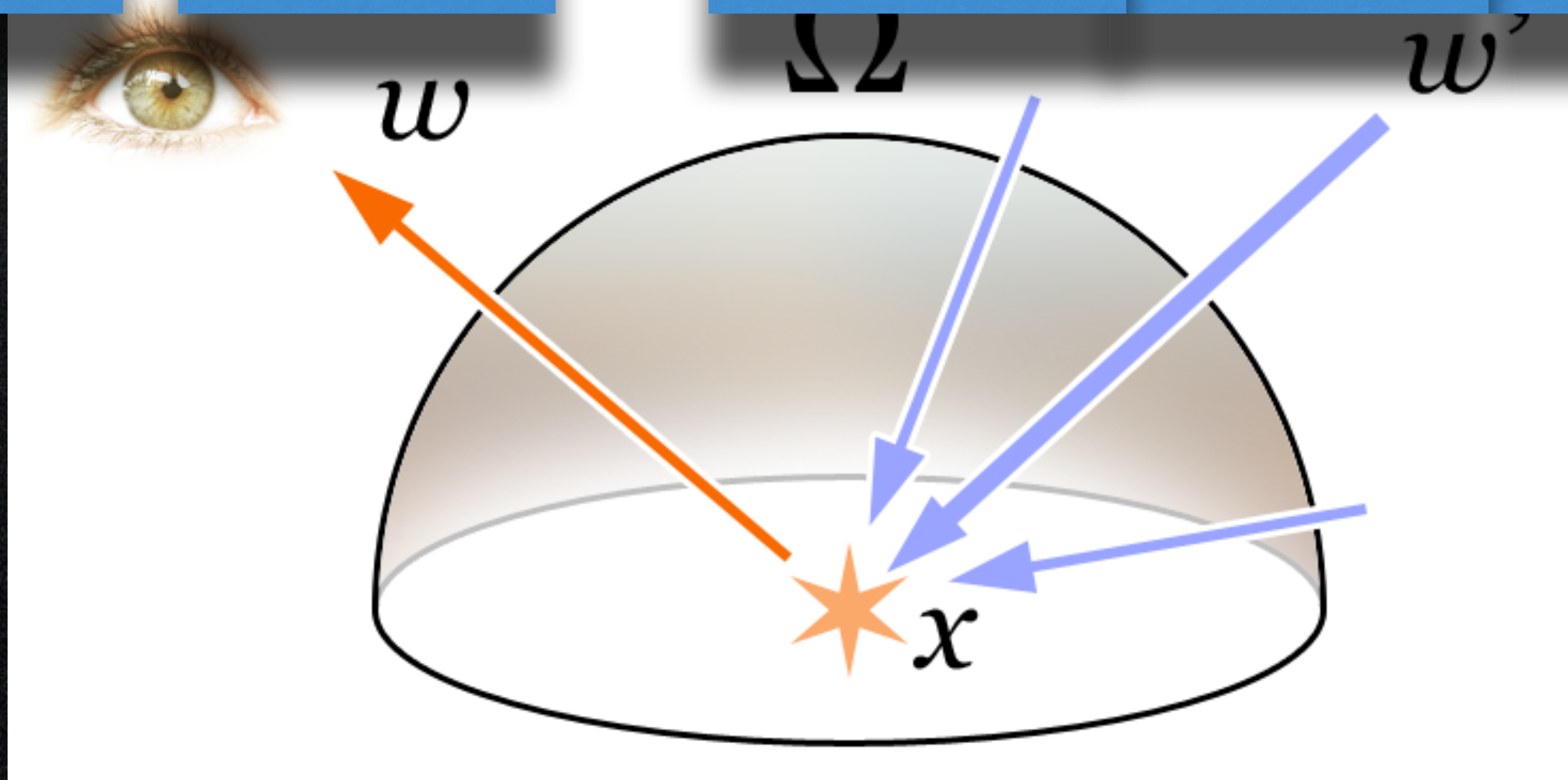
Light out

Emitted light

BRDF

Light in

Atten-uation



Rendering Equation

$$L_o(\mathbf{x}, \omega, \lambda, t) = L_e(\mathbf{x}, \omega, \lambda, t) + \int_\Omega f_r(\mathbf{x}, \omega', \omega, \lambda, t) L_i(\mathbf{x}, \omega', \lambda, t)(-\omega' \cdot \mathbf{n}) d\omega'$$

Multiple bounces means L_o becomes L_i

Rendering Equation

$$L_r(\mathbf{x}, \omega, \lambda, t) = L_e(\mathbf{x}, \omega, \lambda, t) + \int_\Omega f_r(\mathbf{x}, \omega', \omega, \lambda, t) L_r(\mathbf{x}, \omega', \lambda, t)(-\omega' \cdot \mathbf{n}) d\omega'$$

Now we have a recurrence relation

Informal Explanation

$$L_r(\mathbf{x},\omega,\lambda,t) = L_e(\mathbf{x},\omega,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega',\omega,\lambda,t) L_r(\mathbf{x},\omega',\lambda,t)(-\omega'\cdot\mathbf{n})d\omega'$$

Turn the integral into infinite sum

$$L_r(\mathbf{x},\omega,\lambda,t) = L_e(\mathbf{x},\omega,\lambda,t) + \Sigma_{\omega'\in\Omega} f_r(\mathbf{x},\omega',\omega,\lambda,t) L_r(\mathbf{x},\omega',\lambda,t)(-\omega'\cdot\mathbf{n})$$

Informal Explanation

$$L_r(\mathbf{x}, \omega, \lambda, t) = L_e(\mathbf{x}, \omega, \lambda, t) + \Sigma_{\omega' \in \Omega} f_r(\mathbf{x}, \omega', \omega, \lambda, t) L_r(\mathbf{x}, \omega', \lambda, t)(-\omega' \cdot \mathbf{n})$$

Think of the function as vectors of infinite length

$$L_{\mathbf{x},\omega} = E_{\mathbf{x},\omega} + \Sigma_{\omega' \in \Omega}(-\omega' \cdot \mathbf{n}) F_{\mathbf{x},\omega',\omega} L_{\mathbf{x},\omega'}$$

Informal Explanation

$$L_{\mathbf{x},\omega} = E_{\mathbf{x},\omega} + \Sigma_{\omega' \in \Omega}(-\omega' \cdot \mathbf{n}) F_{\mathbf{x},\omega',\omega} L_{\mathbf{x},\omega'}$$

$$L = E + KL$$

$$L - KL = E$$

$$(I - K)L = E$$

$$L = (I - K)^{-1} E$$

$$L = TE$$ Our familiar form

Significance

$$T = (I - K)^{-1}$$

BRDF + attenuation rolled together

$$(I - K)^{-1} = I + K + K^2 ...$$

Emission Direct 1st bounce

$$L = E + KE + K^2 E ...$$

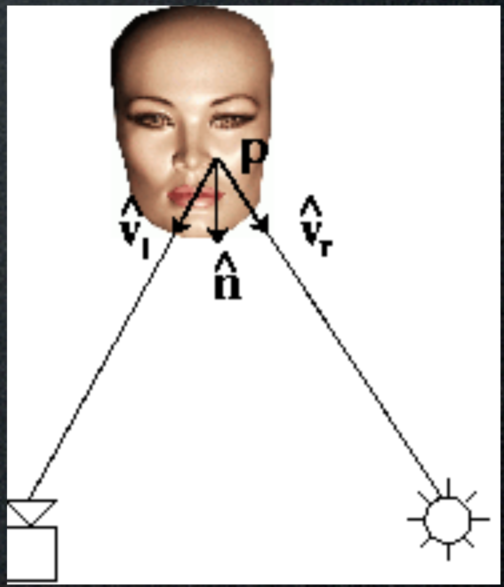
Concrete Example

Exitant light (camera)

$$C = TL$$

A 4D slice (no variation in direction)

Incident light (projector)



Scene Relighting

We've already acquired T

Plug in a novel illumination

$$C' = TL'$$

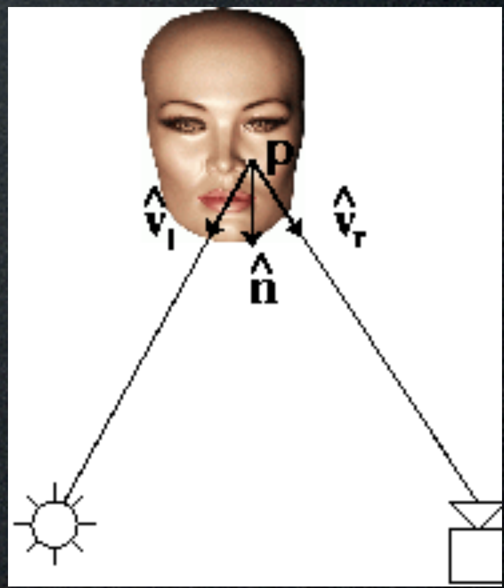
And we have the relit scene

Helmholtz Reciprocity

Swap camera and illumination

$$C'' = T^T L''$$

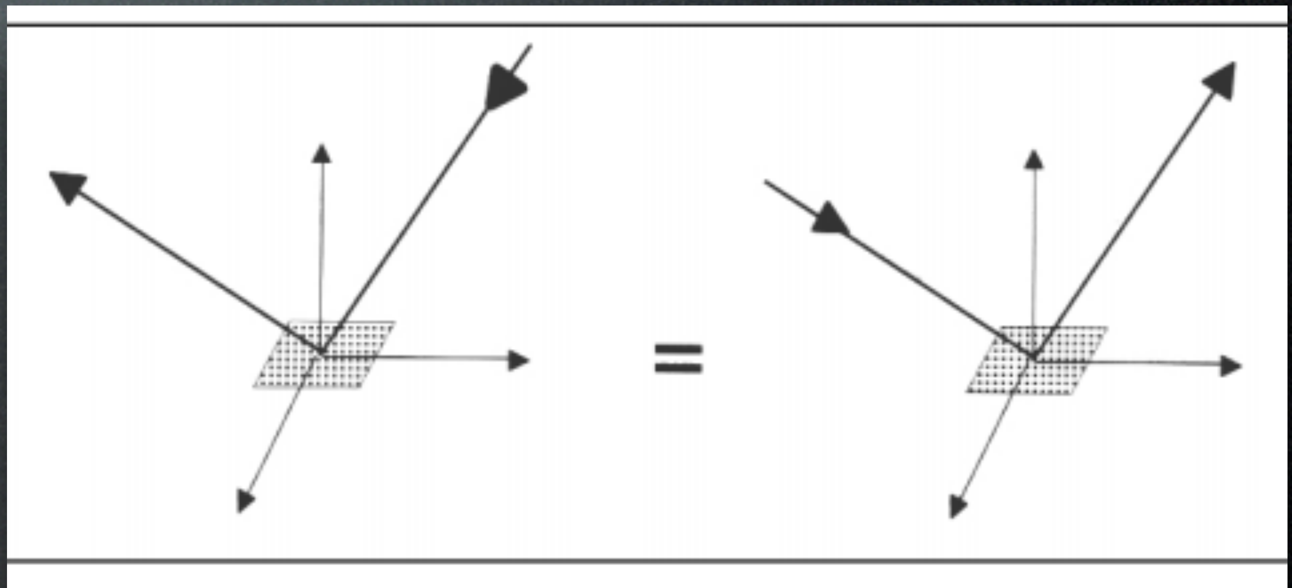
Just transpose light transport



Note: $T^T \neq T^{-1}$ due to light absorption and scattering

When does it hold?

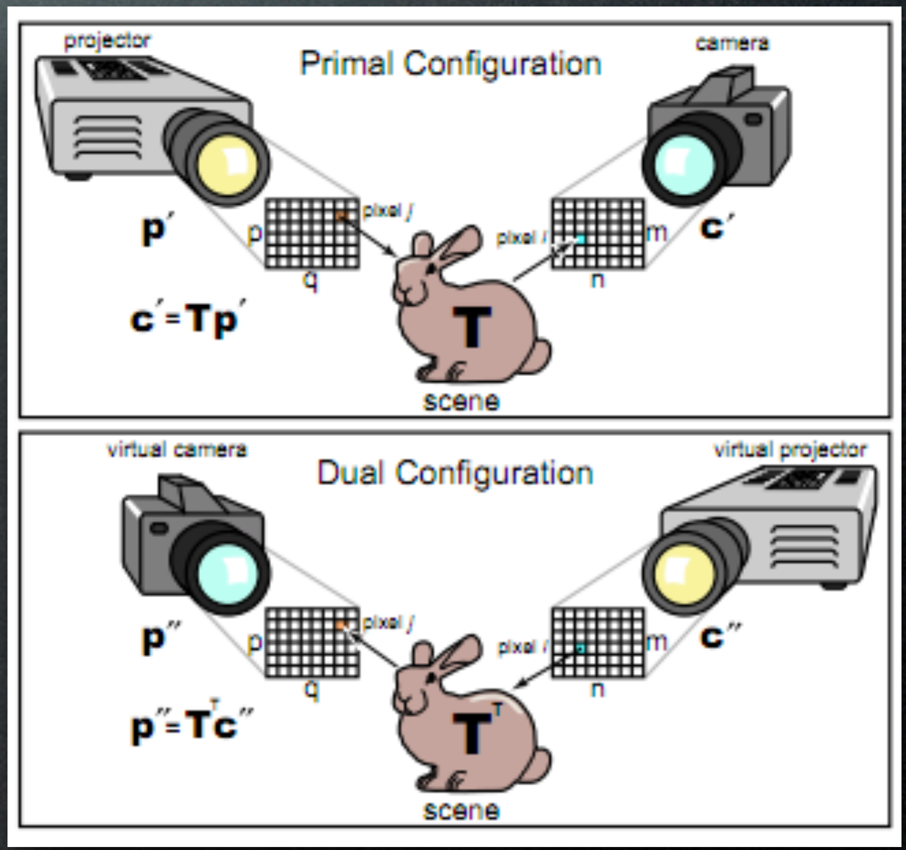
- When the BRDF is symmetric (swap incident and reflected directions)
- Enforced



Dual Photography

Pradeep Sen et al.

- Capture T
- Synthesize projector's view with Helmholtz reciprocity



Acquiring T

L

0.7 = 0.7 1

C T

- Example: 2D projector - photosensor
- T is 1-by-mn where projector resolution is m-by-n
- Brute force approach

Acquiring T

L

0.5 = 0.7 0.5 1

C T

- Example: 2D projector - 1D photosensor
- T is 1-by-mn where projector resolution is m-by-n
- Brute force approach

Until you have T

Relighting

L

1
1
1

1.3 = 0.7 0.5 0.3

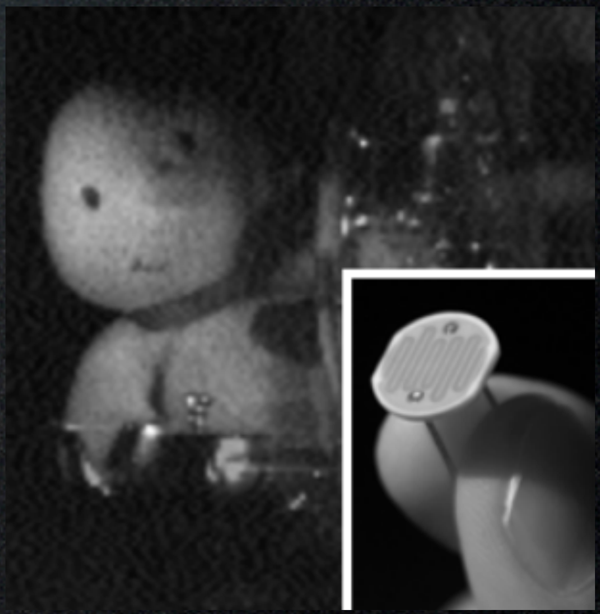
C T

- Specify novel I

Dual Photo

- Apply Helmholtz reciprocity
- Projector -> camera
- Photosensor -> point light source

$$\underset{C'}{\begin{bmatrix}0.7\\0.5\\0.3\\ \vdots\end{bmatrix}} = \underset{T^T}{\begin{bmatrix}0.7\\0.5\\0.3\\ \vdots\end{bmatrix}} \underset{L'}{\begin{bmatrix}1\end{bmatrix}}$$

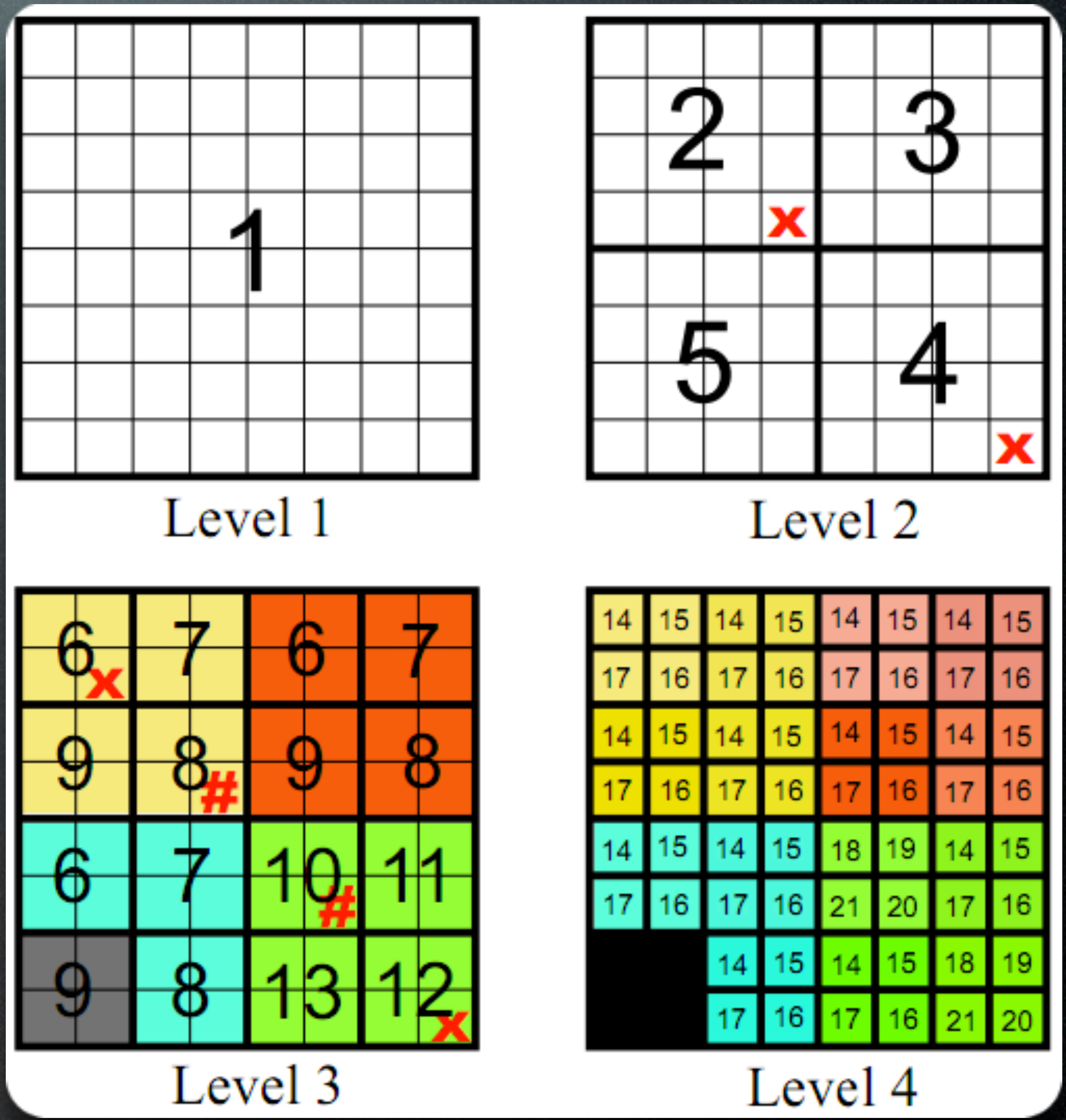


Efficiency

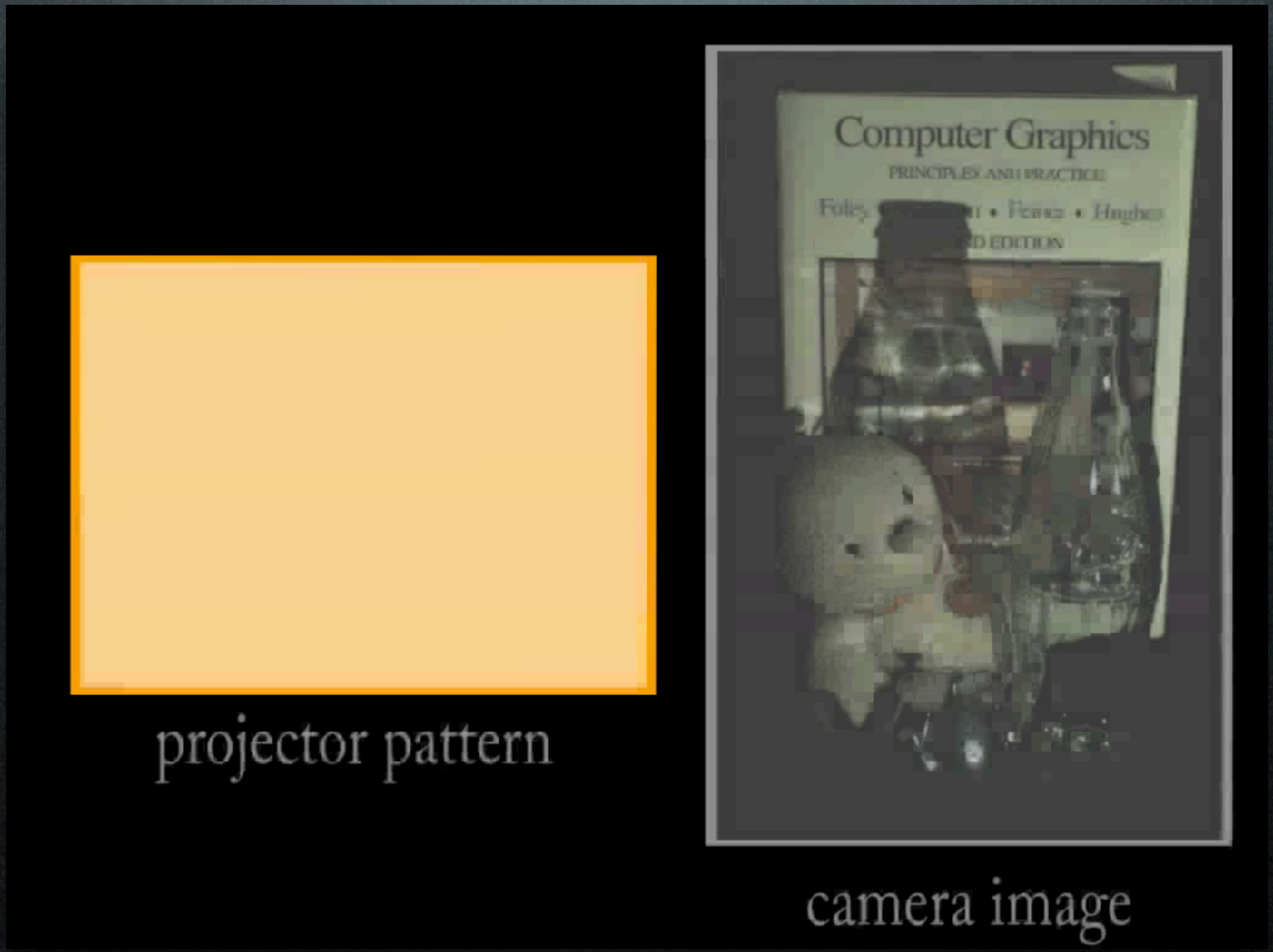
- For m-by-n projector and p-by-q camera brute force approach
  - mn images
  - 15 megapixel camera, VGA projector, 24 bit color depth = ~12.5 TB per scene (no compression)
- Assuming 1 sec per image (exposure, storage, processing) ~85 hours to acquire
- *Multiplexing approach is necessary*

Adaptive Multiplexed Illumination

- Subdivide illumination space
- Check for conflicts in camera space
- If no collision, measure with both illuminations and sort out the separation later
- Degrades to brute force in complex scenes



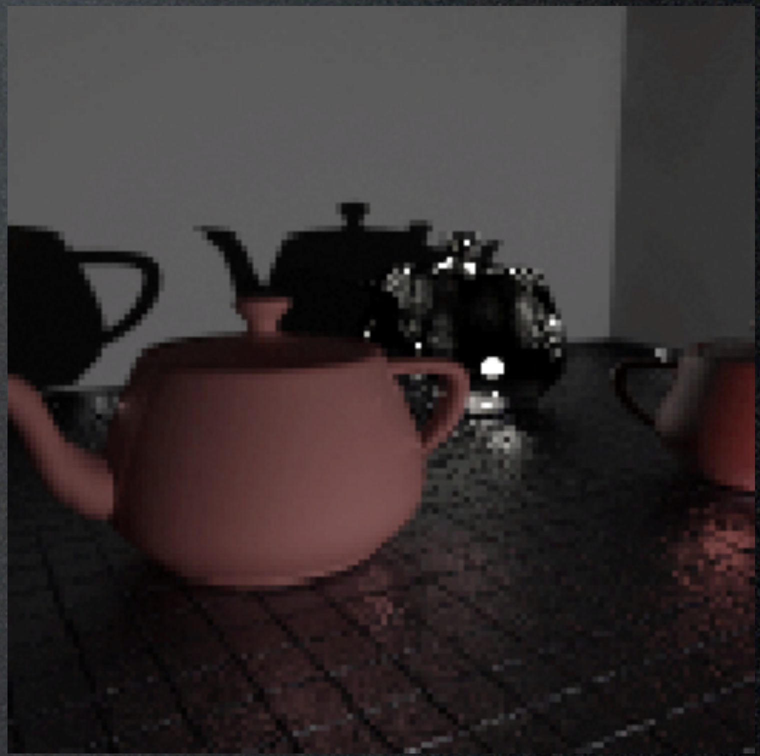
Example


projector pattern
Computer Graphics
PRINCIPLES AND PRACTICE
camera image

Example

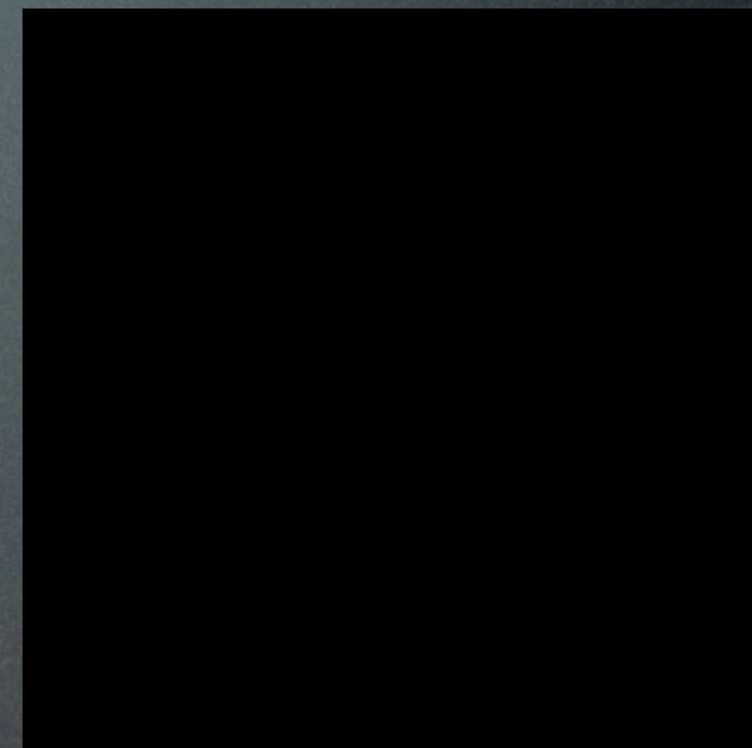
Personal Experiment

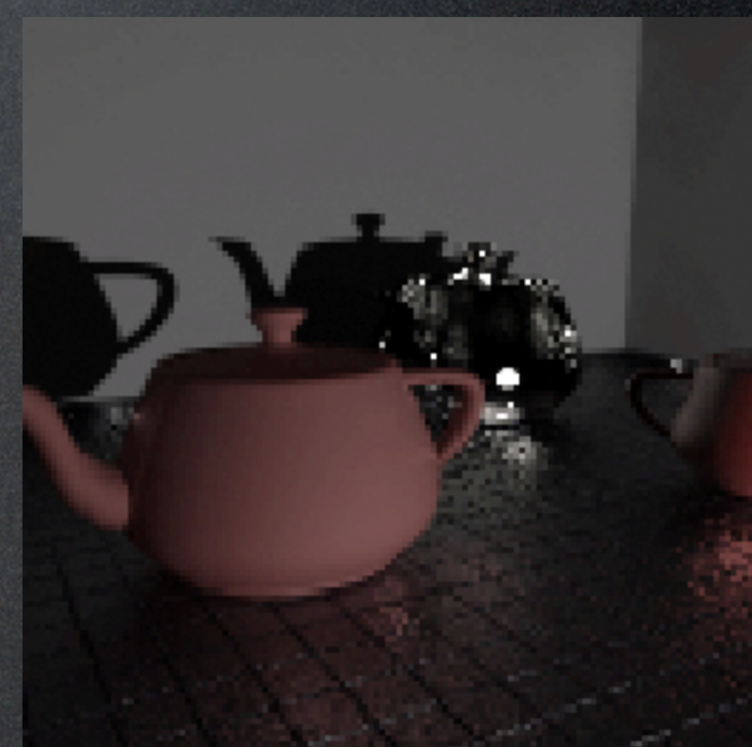
- Rendered 128x128 images with 128x128 projector
- Used brute force approach
- Performed relighting and dual photography
- Artifacts present in dual image (Renderer light sampling?)



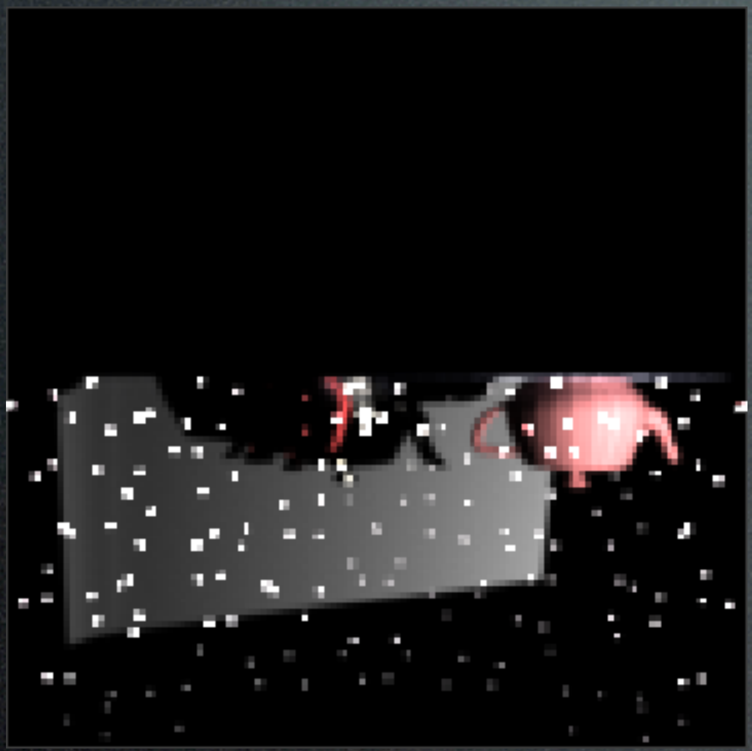
Personal Experiment

- Rendered 128x128 images with 128x128 projector
- Used brute force approach
- Performed relighting and dual photography
- Artifacts present in dual image (Renderer light sampling?)



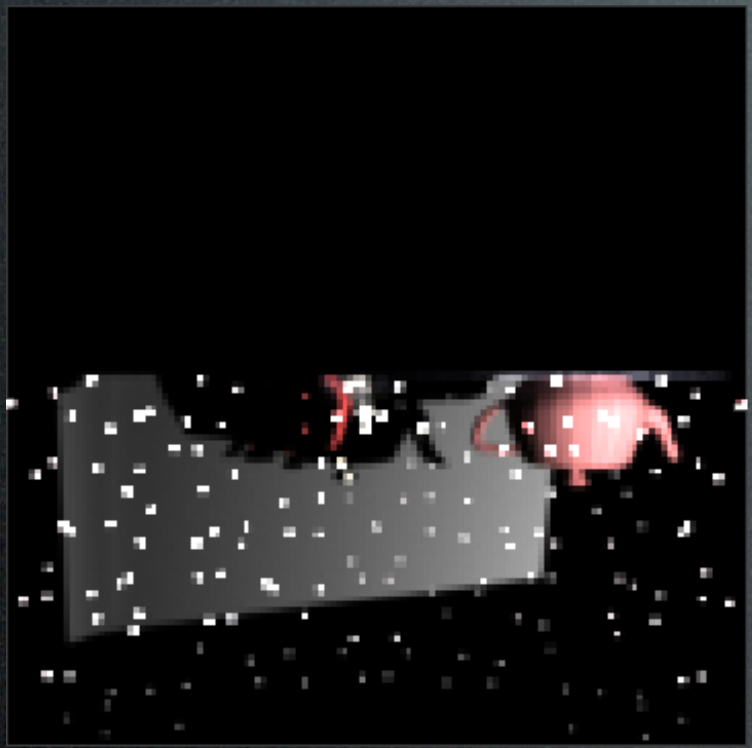


Personal Experiment



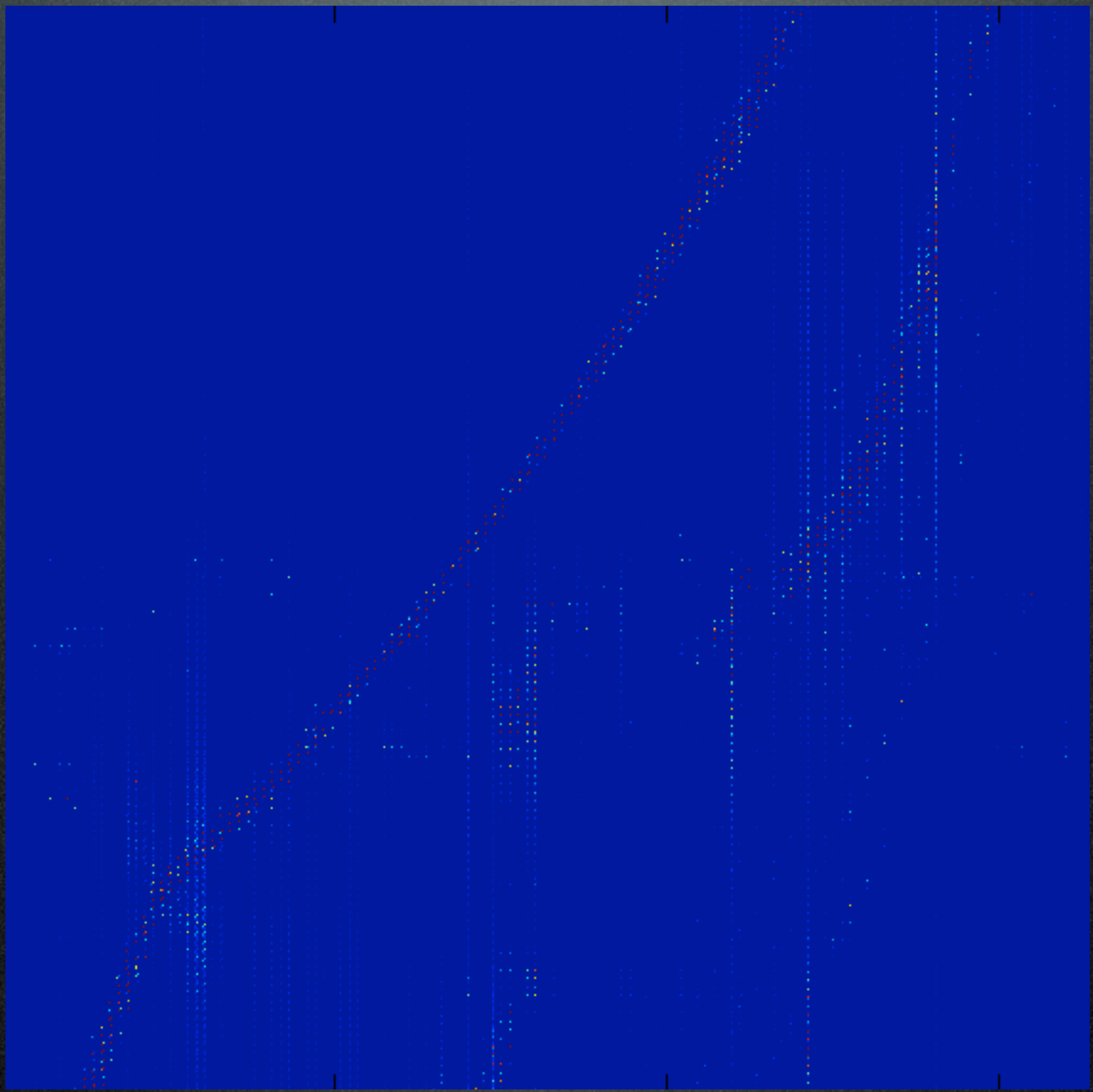
Projector must have been upside down...oops

Personal Experiment



Projector must have been upside down...oops

Sparsity Observation



T is data sparse

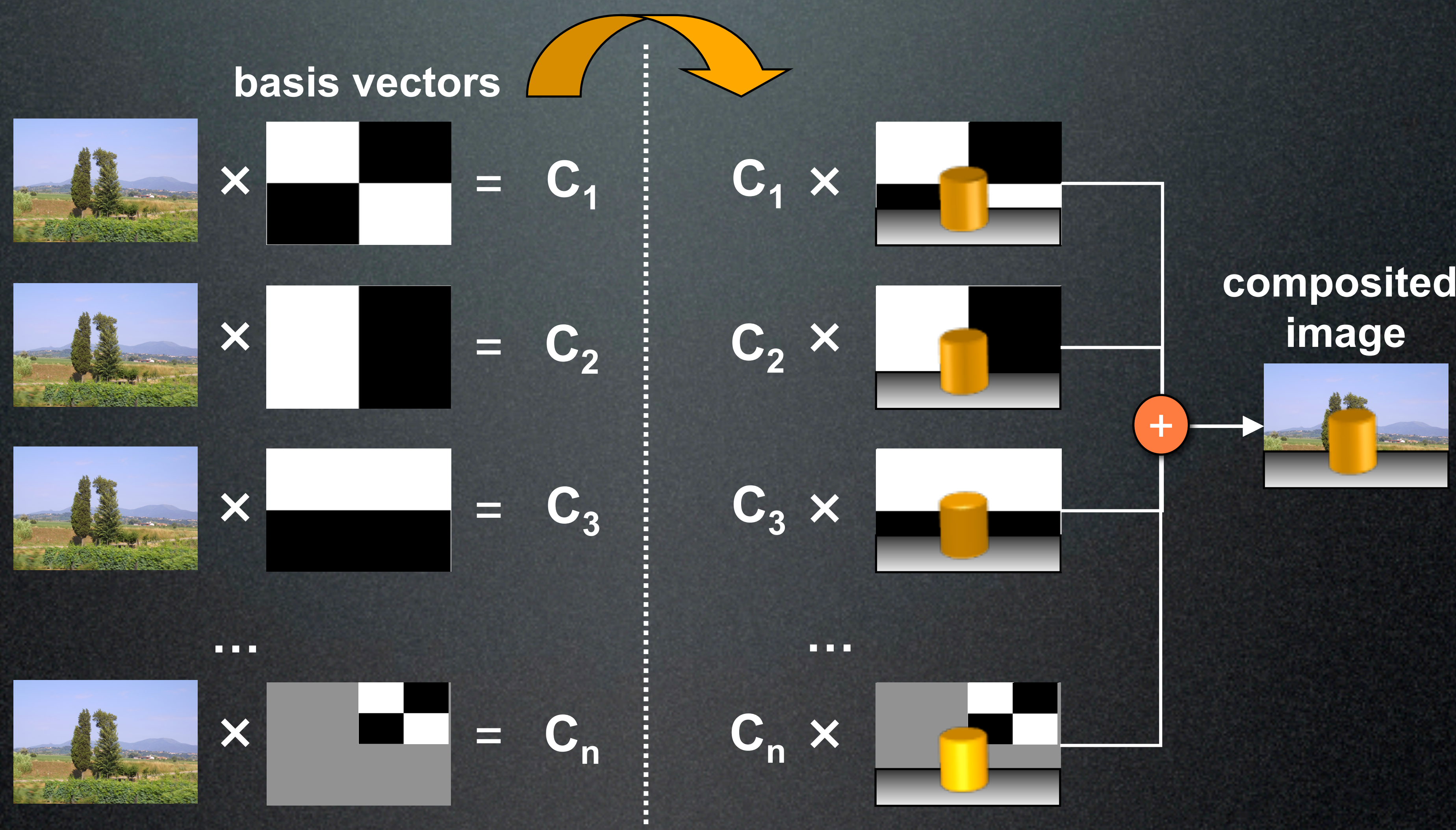
- For m-by-n projector and p-by-q camera, the 4D slice still takes O(mnpq) bytes to store (assuming no compression)
- Let's find ways to exploit data sparsity
  - Sparse entries
  - Low rank approximations
  - Compressible basis transformations

Next Question

- Is the light transport data-sparse even more so in another basis?
- Idea: Let's try a basis used for image compression - wavelets
- Why wavelets? Why not just a Fourier basis?
- Localized in both frequency and space

Wavelet Environment Matting

Pieter Peers et al.



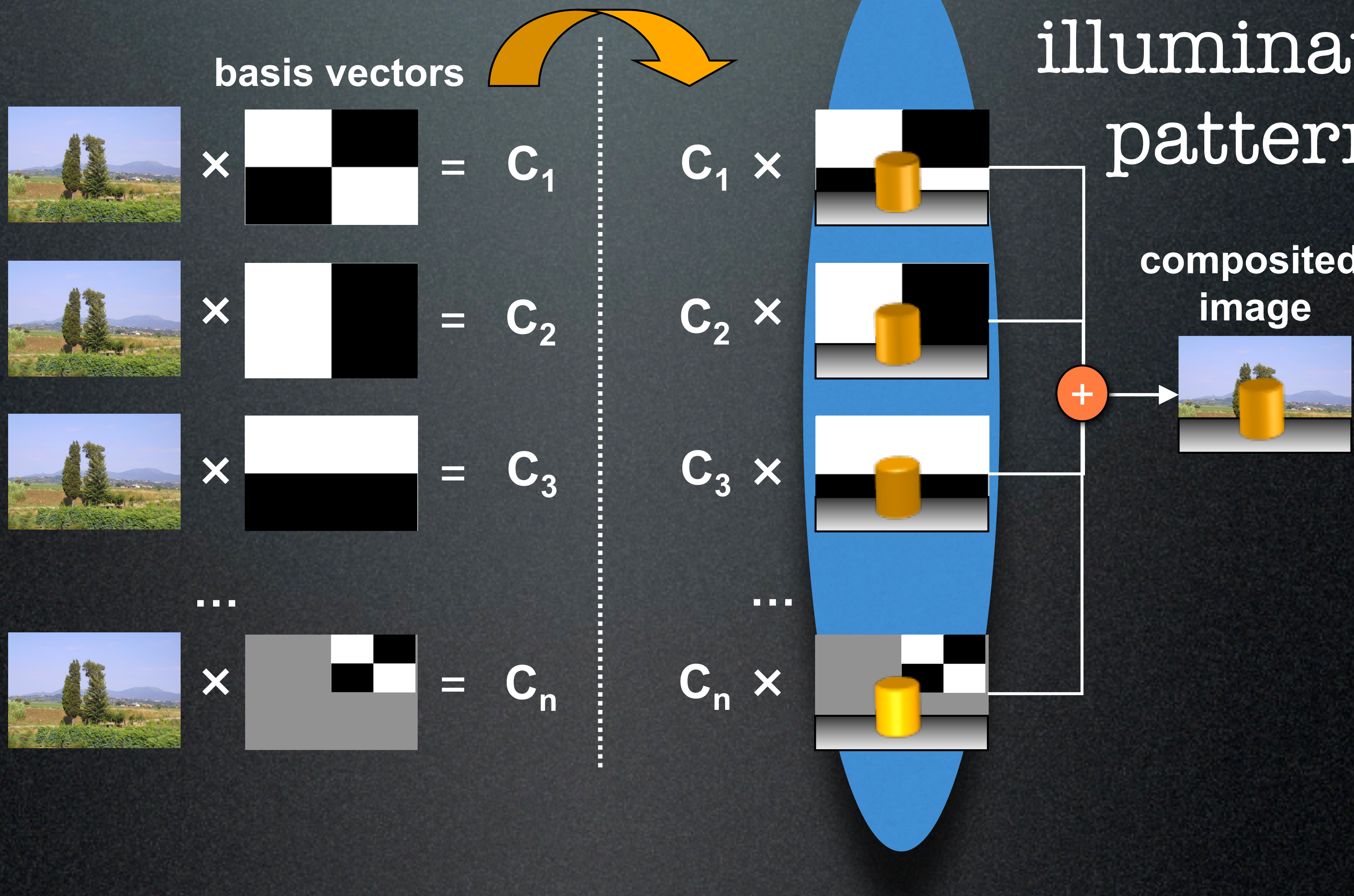
How it works

- Brute force: C = TI (columns of identity form single pixel patterns)
- Turn it into $C=T(BB^T)I$
- $C=(TB)(B^TI)$
- $C=(TB)B^T$

What does that mean?

$C=(TB)B^T$

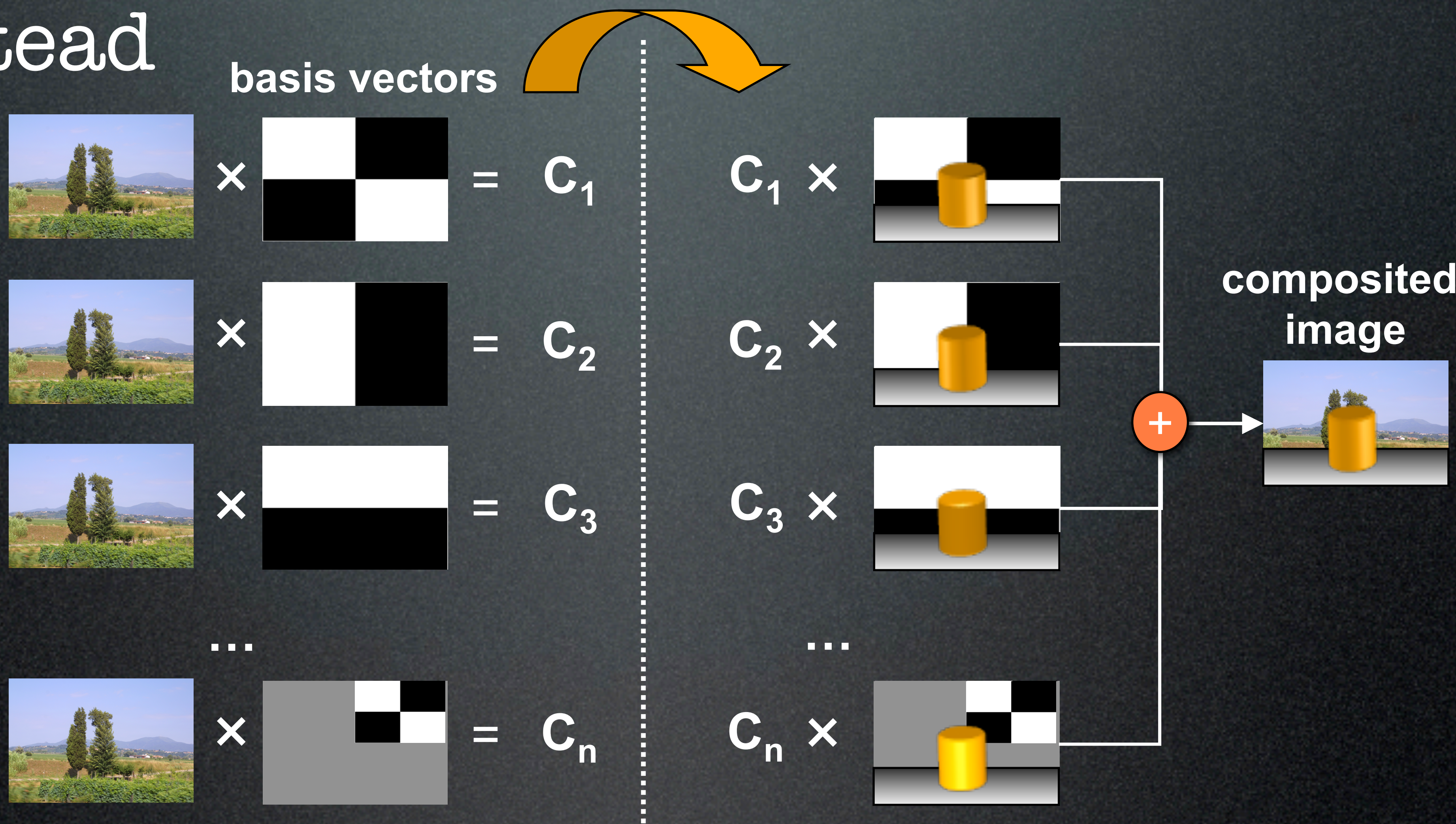
Use vectors of basis as illumination patterns



What does that mean?

$C=(TB)B^T$

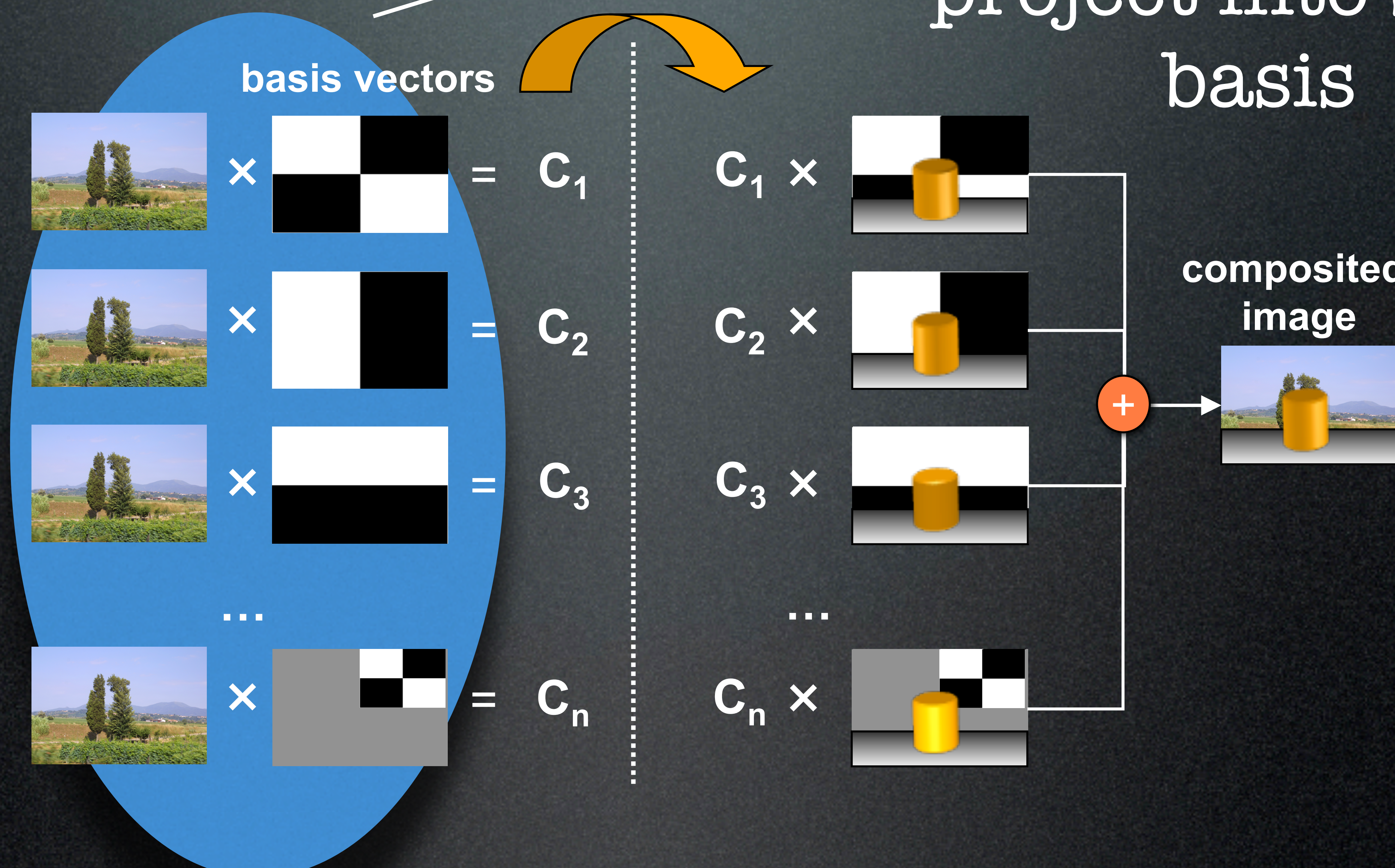
And measure T projected onto B instead



What does that mean?

$C' = (TB)(l'^T B)^T$

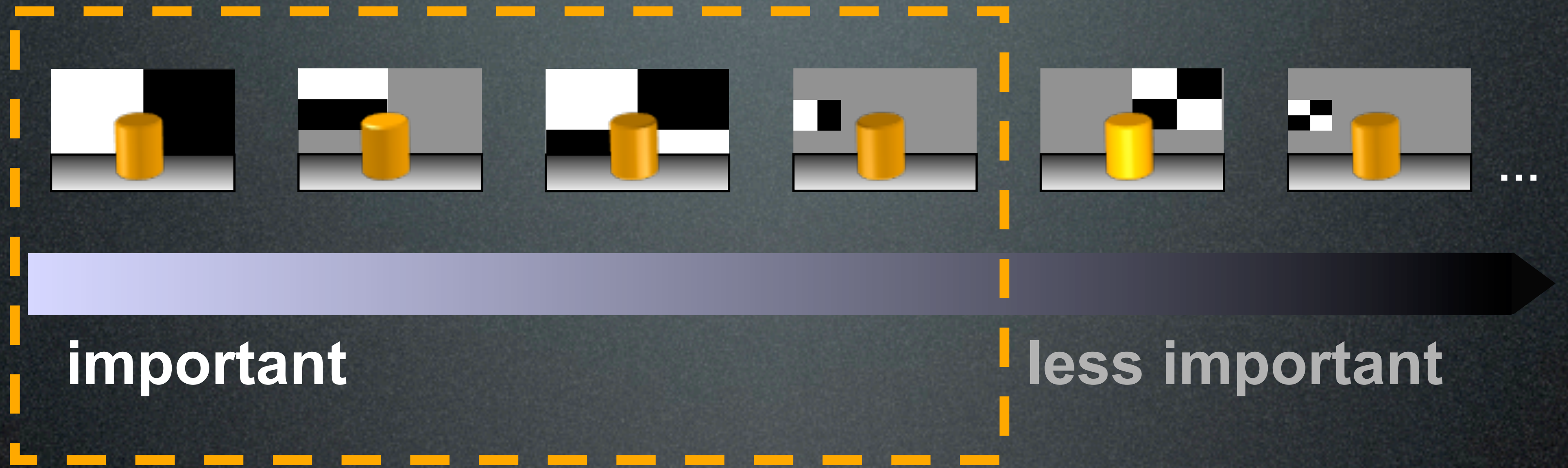
So if we want a novel illumination, project into same basis



Why do we care?

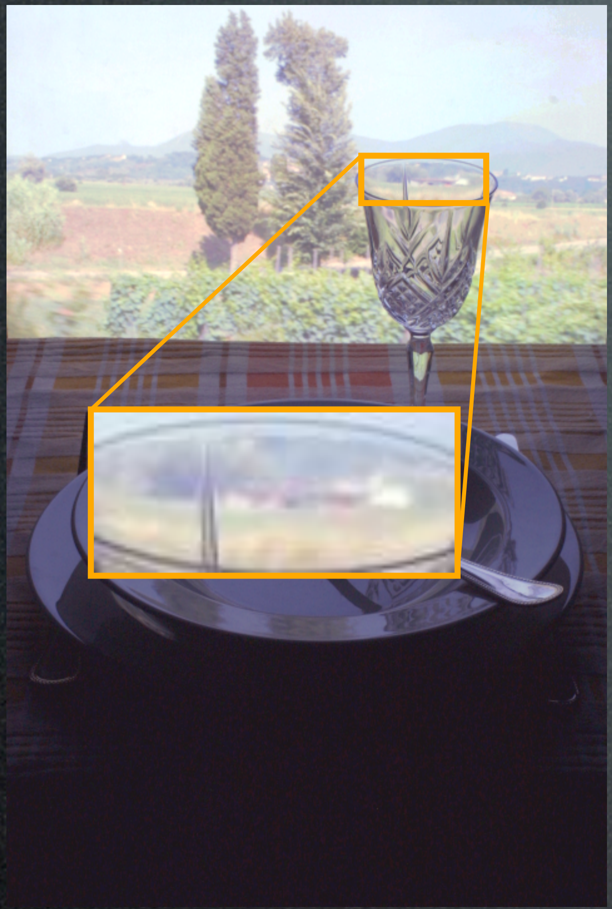
- At first glance, same number of captures as brute force
- More than 1 pixel illuminated - better SNR
- Under-sampling
  - Wavelets exploit spatial relationships
  - Might be better than discarding illumination pixels

Choosing wavelet basis vectors

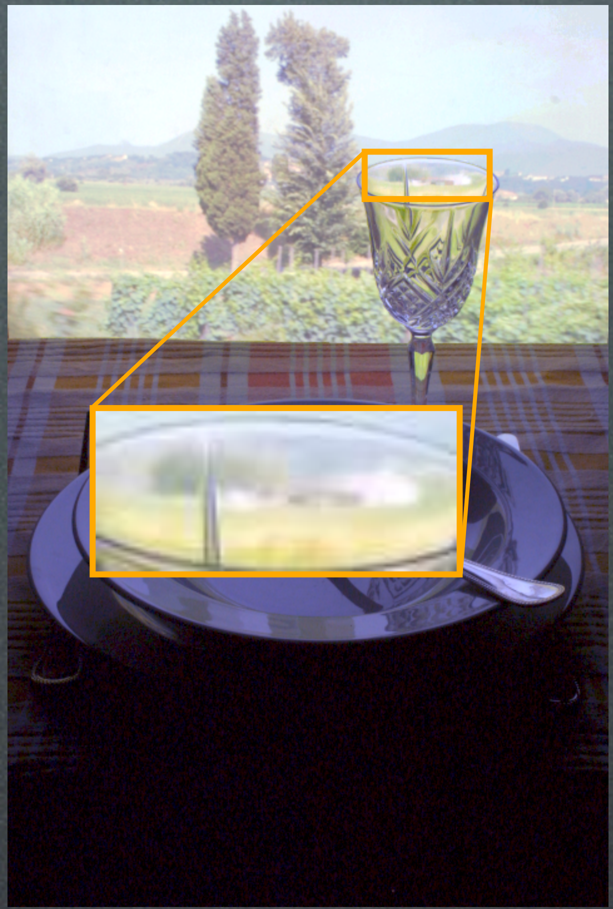


Uses a feedback algorithm to choose next best vector

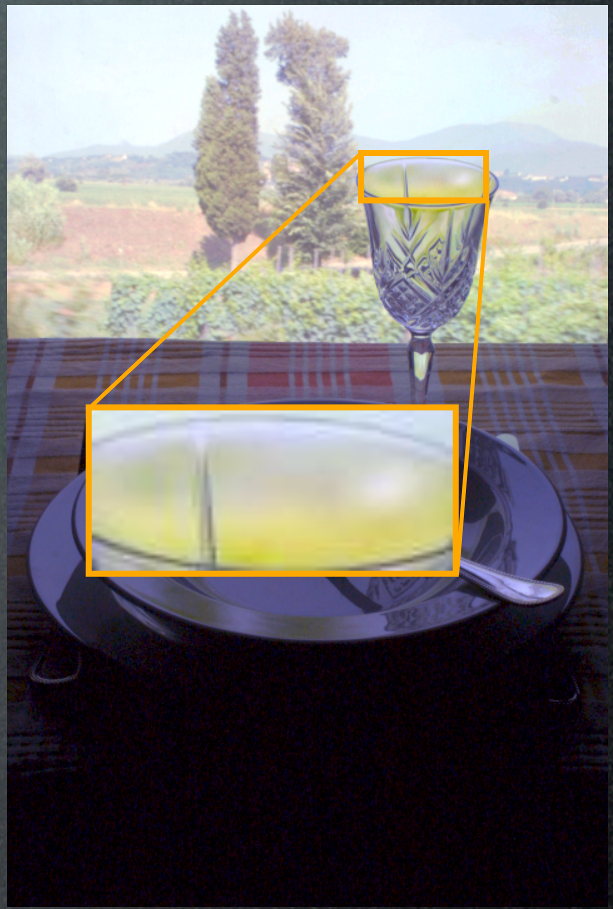
Results



Reference image



1000 Haar patterns



1000 Daubechies (9,7) patterns

Video Results

Video Results

Main take-away

- Haar wavelet basis is good for measuring T
  - Or at least they contrived their test scenes well enough to be convincing

Compressive Light Transport Sensing

Pieter Peers et al.

- Bypass this whole wavelet basis vector selection and use compressive sensing theory
- We had $C=T(BB^T)I$ where B is the Haar wavelet basis
- In reality we didn't use all of I
- $C=(TB)(B^TA)$ where A is subset of I's columns

How it works

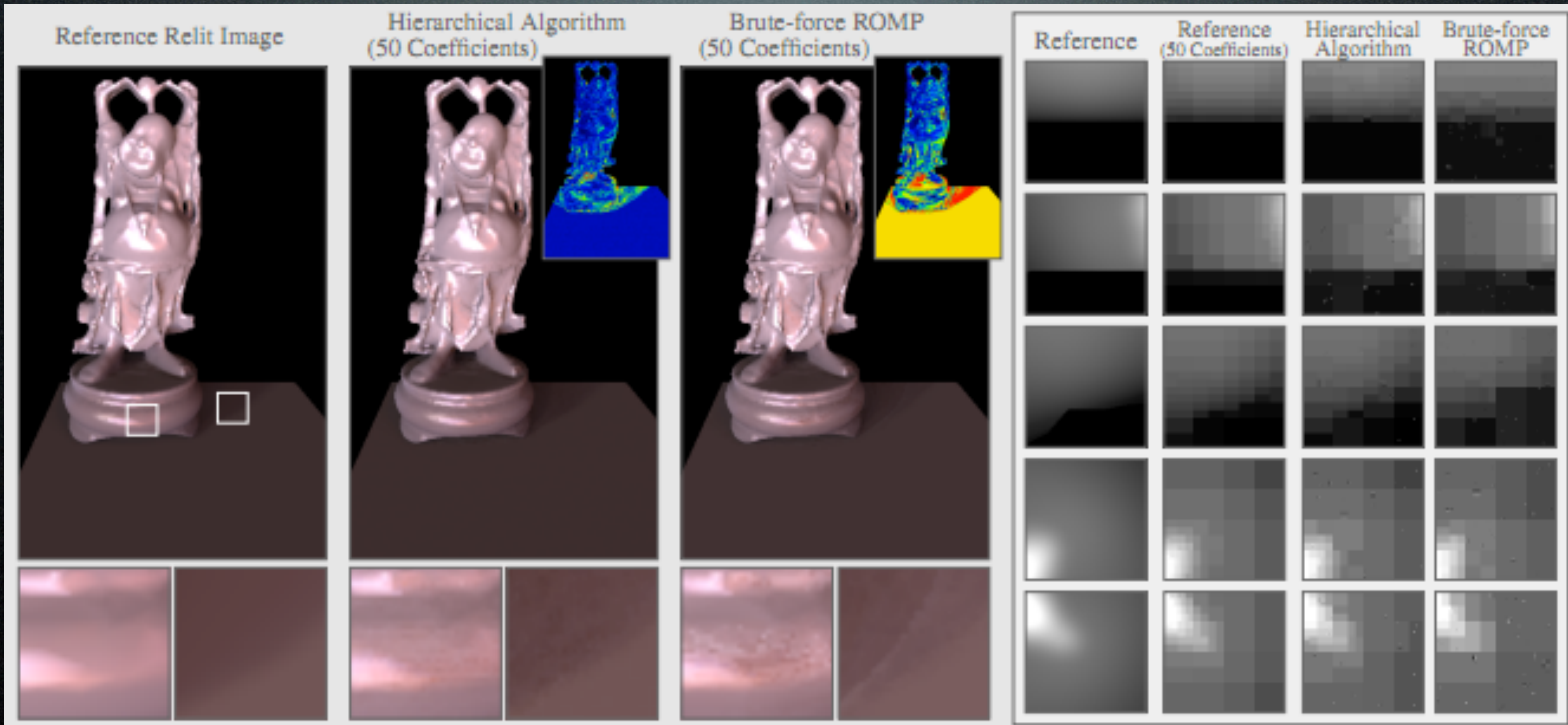
- Measure rows of T one at a time
- $c_i = (t_i B)(B^T A)$
- $c_i^T = (A^T B)(B^T t_i^T)$

Row of T projected onto wavelet basis

- Last paper showed empirically is sparse
- CS theory applies

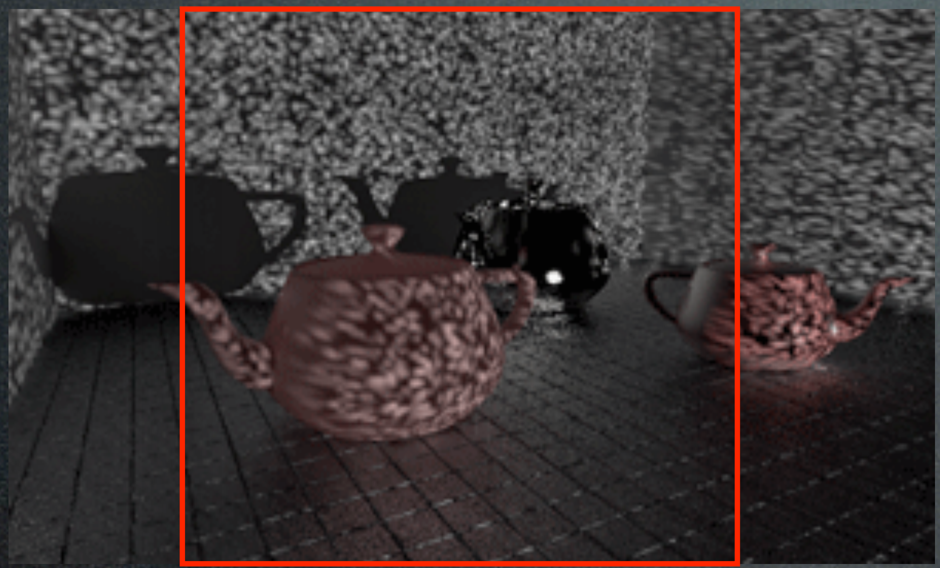
CS Theory - High level

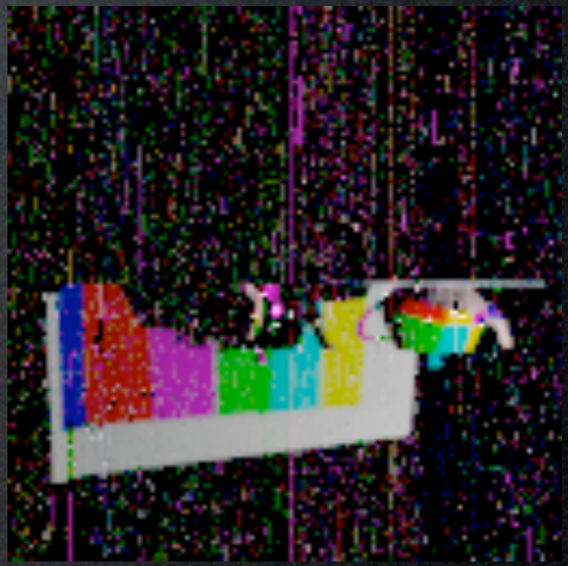
- Ignoring important properties like how to select A...
- $c_i^T = (A^T B)(B^T t_i^T) \Rightarrow y = \phi^T x$
- But x is sparse
- Want to solve $\mathrm{argmin}_x \, ||x||_0$ s.t. $y = \phi^T x$
- NP-complete
- Settle for $\mathrm{argmin}_x \, ||x||_1$ s.t. $y = \phi^T x$
- Linear programs - no strongly polynomial algorithm known
- Basis pursuit, orthogonal matching pursuit, ROMP, CoSaMP, etc.


Reference Relit Image
Hierarchical Algorithm
(50 Coefficients)
Brute-force ROMP
(50 Coefficients)
Reference
Reference
(50 Coefficients)
Hierarchical
Algorithm
Brute-force
ROMP

Results

Compressive Sensing Experiment





150 samples
20 wavelet coefficients

One last idea

- All these methods acquire T
- Can we compute without acquiring T directly?

Optical Computing for Fast Light Transport Analysis

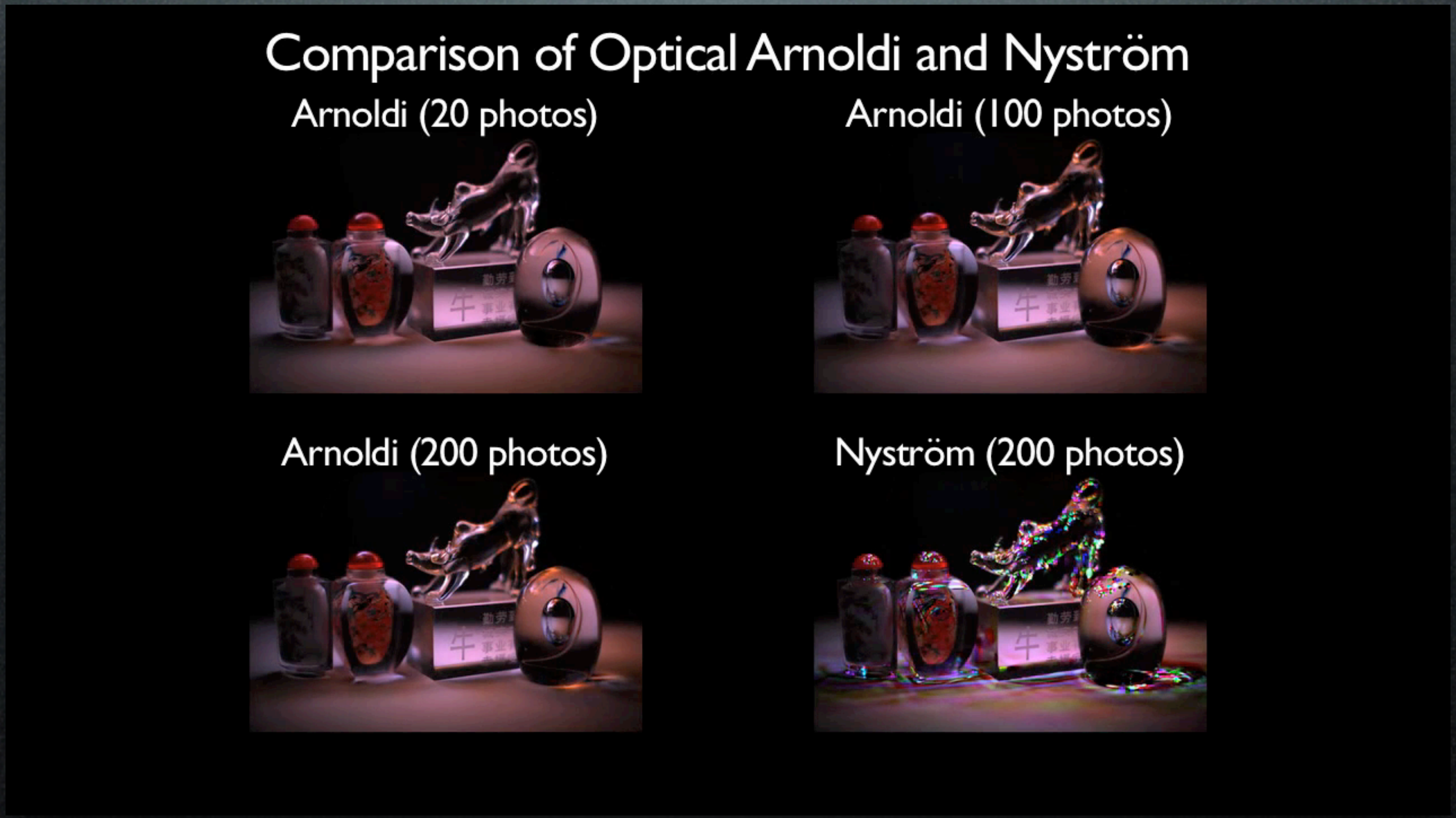
O'Toole et al.

- Krylov subspace methods
  - Arnoldi
  - GMRES
- Wherever you see Tl or $T^T l$, replace with black box physical process

Algorithm	Numerical objective	Step 1	Step 4	Step 5
Power iteration (Section 2.1)	estimate principal eigenvector of $\mathbf{T}$	$\mathbf{l}_1$ = positive vector	$\mathbf{l}_{k+1} = \mathbf{p}_k / \lVert \mathbf{p}_k \rVert_2$	**return** $\mathbf{l}_{k+1}$
Arnoldi (Section 3)	compute rank-K approximation of $\mathbf{T}$	$\mathbf{l}_1$ = non-zero vector	$\mathbf{l}_{k+1} = \mathrm{ortho}(\mathbf{l}_1, \ldots, \mathbf{l}_k, \mathbf{p}_k)$ $\mathbf{l}_{k+1} = \mathbf{l}_{k+1} / \lVert \mathbf{l}_{k+1} \rVert_2$	**return** $[\mathbf{p}_1 \cdots \mathbf{p}_K][\mathbf{l}_1 \cdots \mathbf{l}_K]^\dagger$
Generalized minimal residual (Section 4)	find vector $\mathbf{l}$ such that $\mathbf{p} = \mathbf{T}\,\mathbf{l}$	$\mathbf{l}_1$ = target photo $\mathbf{p}$	$\mathbf{l}_{k+1} = \mathrm{ortho}(\mathbf{l}_1, \ldots, \mathbf{l}_k, \mathbf{p}_k)$ $\mathbf{l}_{k+1} = \mathbf{l}_{k+1} / \lVert \mathbf{l}_{k+1} \rVert_2$	**return** $[\mathbf{l}_1 \cdots \mathbf{l}_K][\mathbf{p}_1 \cdots \mathbf{p}_K]^+ \mathbf{p}$

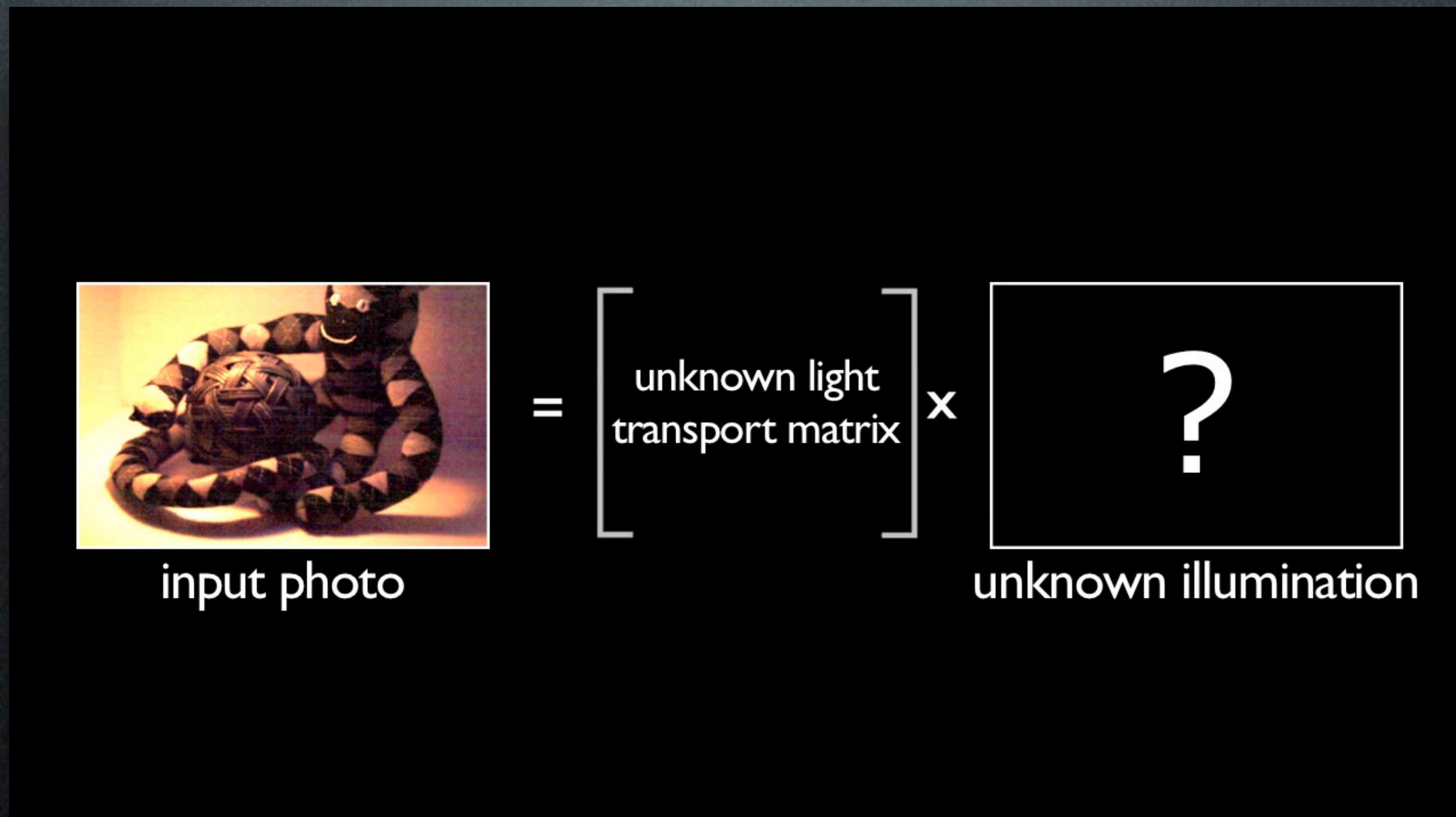
Examples

Optical Arnoldi Results



Optical Arnoldi Results

Optical GMRES Results



Optical GMRES Results

Power Iteration Example

Questions?

Exploit Symmetry

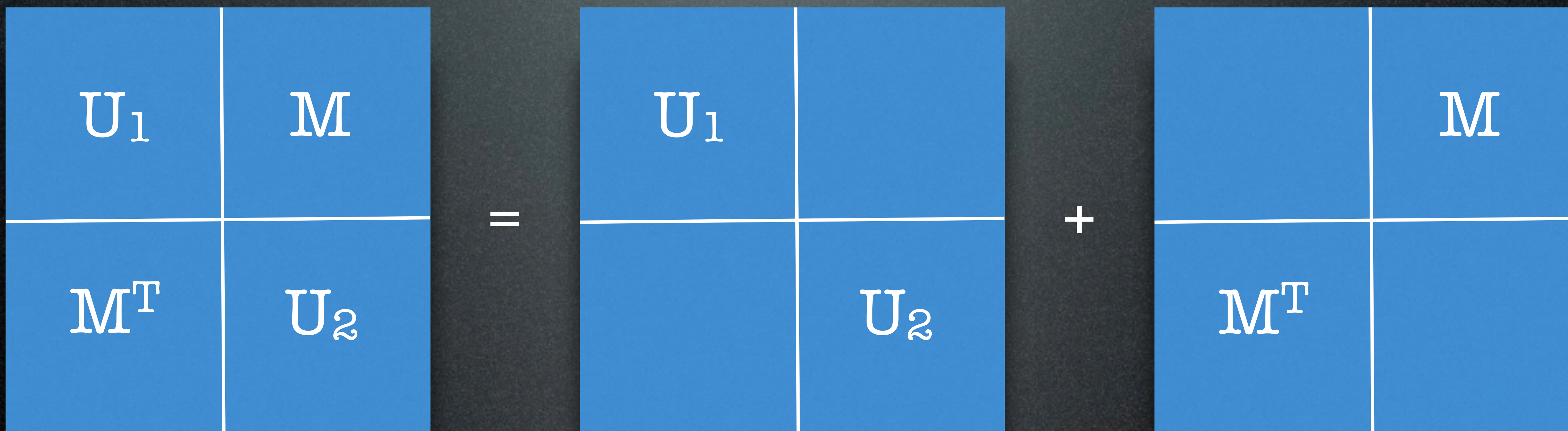
- So far
  - Considered a slice of 8D reflectance field
  - Point camera, 2D projector - 2D slice
  - 2D camera, 2D projector - 4D slice
- Full 8D reflectance field is symmetric
  - From Helmholtz reciprocity

Symmetric Photography

Gaurav Garg et al.

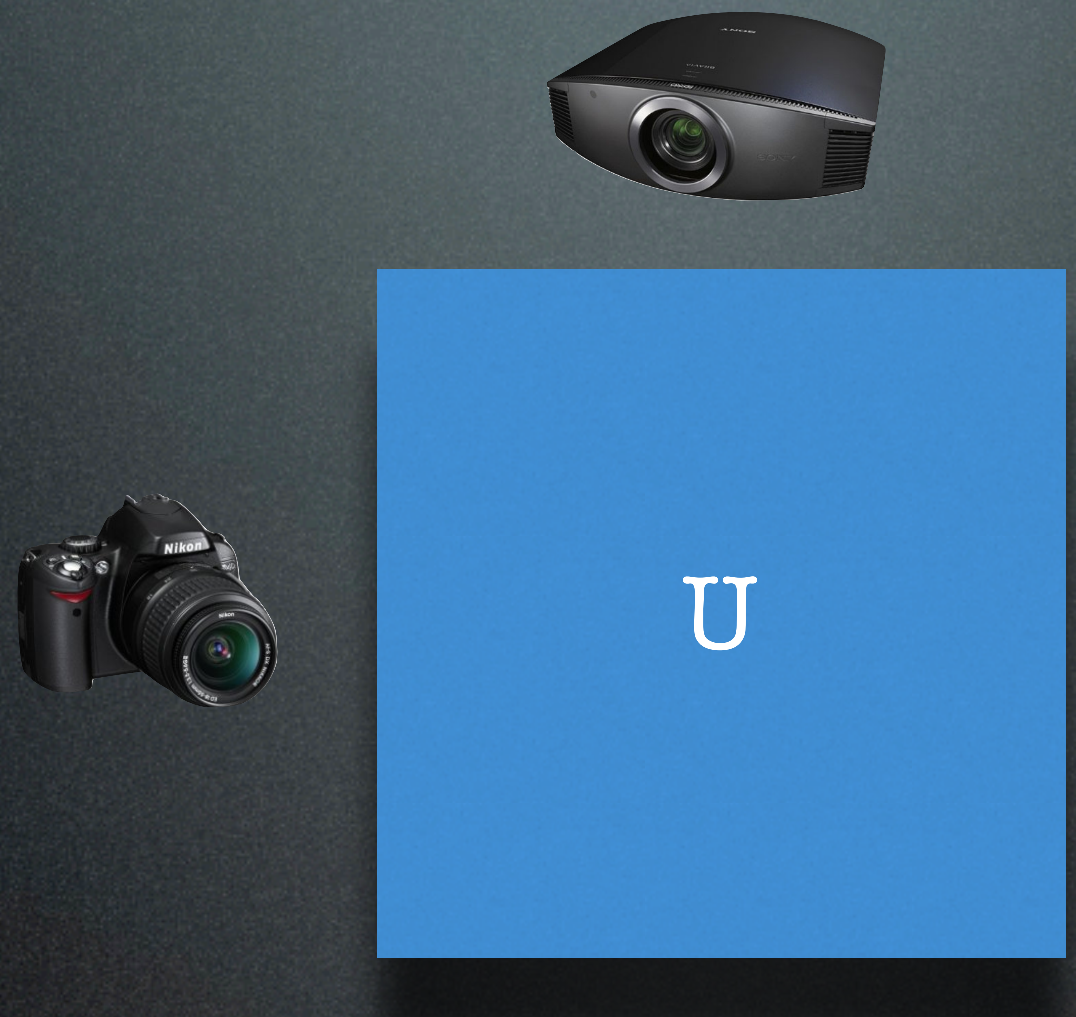
- Key idea: Represent T as hierarchical tensor
- i.e. Don't flatten into giant matrix...preserve locality
- Leaf nodes are rank-1
- Apply previous approaches to higher rank components

Illustration - 2D version



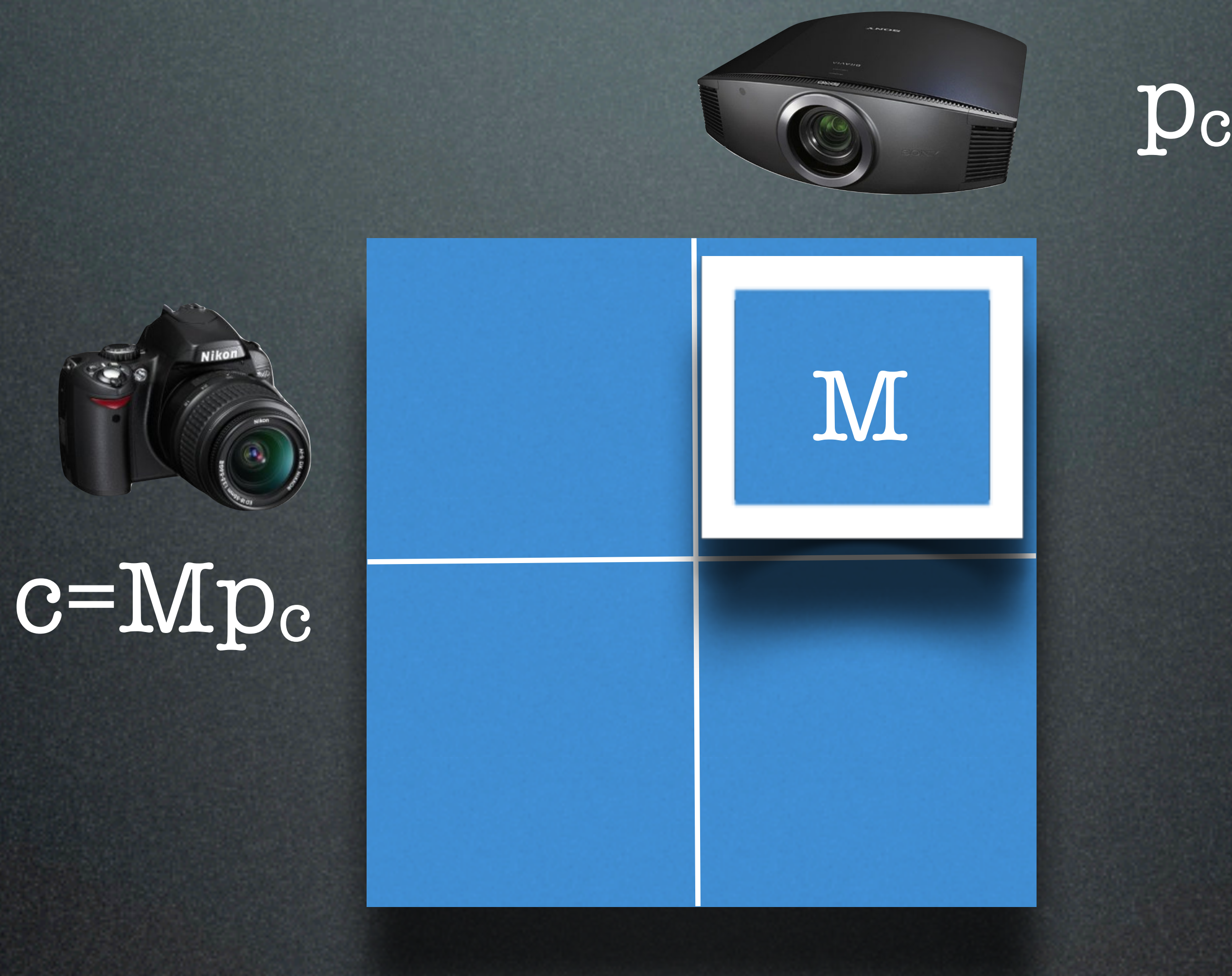
Idea - Measure flood light pattern, then subtract off-diag blocks

Illustration - 2D version



Illuminate all in parallel

Illustration - 2D version



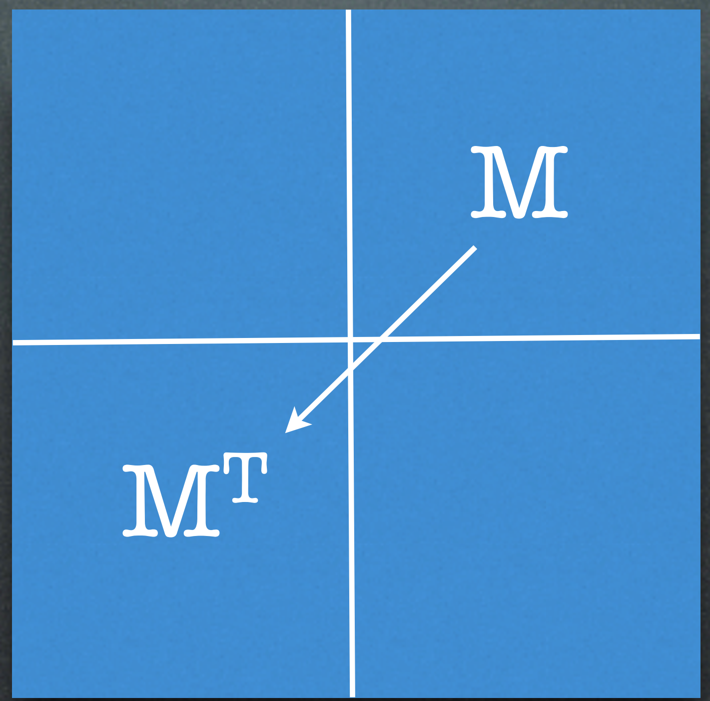
Then decide if the off-diag block is rank-1

Illustration - 2D version

Next Steps

- Choose p_c and p_r to be the 1-vector (to sum cols and rows of M)
- Tensor product of r and c form M
- Check RMS error of low rank approximation
- If below threshold, label as leaf
- Else recurse

Illustration - 2D version

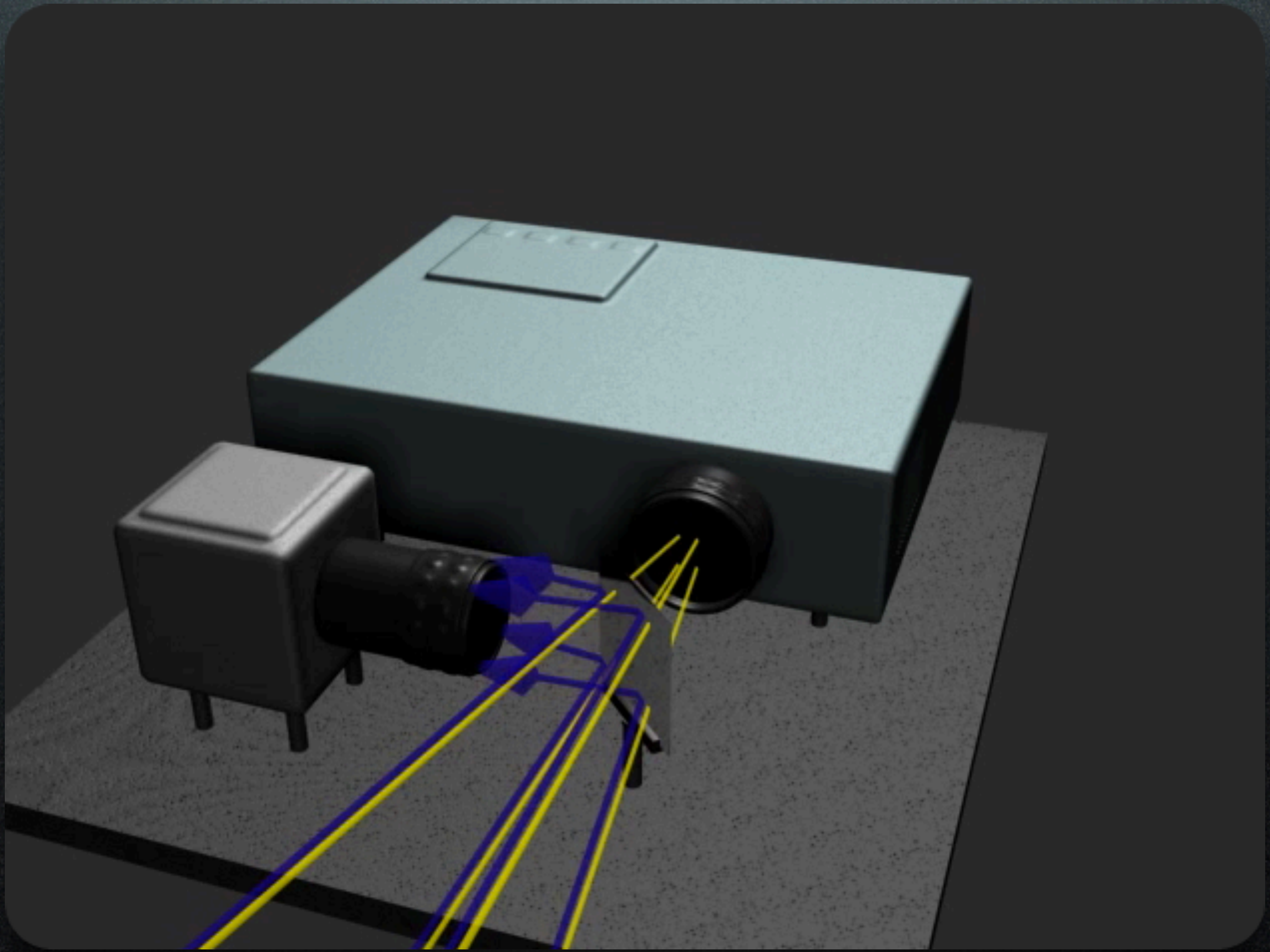


Now we need the diagonal blocks

Illustration - 2D version

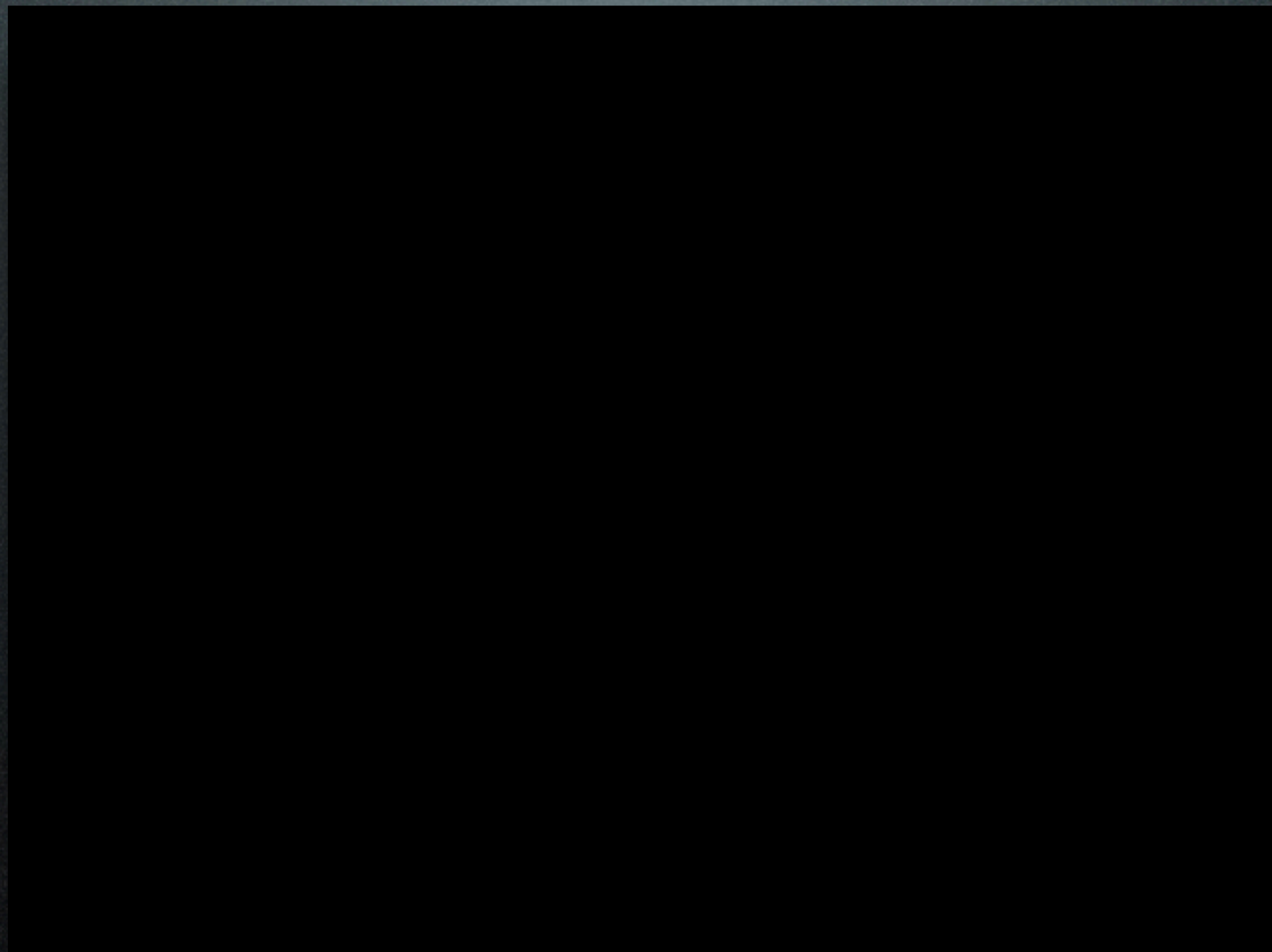
$$\begin{pmatrix} U_1 & M \\ M^T & U_2 \end{pmatrix} = \begin{pmatrix} U_1 & \\ & U_2 \end{pmatrix} + \begin{pmatrix} & M \\ M^T & \end{pmatrix}$$

We now can subtract off off-diag blocks ... and divide and conquer on U_1 and U_2



Experimental Apparatus

Relighting Example



Relighting Example

Comparison to Sen

- Garg captures full 8D reflectance field
- Sen captured up to 6D slices
- Sen degrades to single pixel illumination
- Garg does as well, but exploits data-sparsity to prevent it (better SNR)
- Garg quality degrades with projector-camera misalignment - bit blurry
- Qualitatively, all looks the same to me